A DESIGN OF AN E-COMMERCE RECOMMENDER SYSTEM USING DATA ANALYTICS

THESIS

Submitted in fulfillment of the requirement of degree of

DOCTOR OF PHILOSOPHY

to

The Faculty of Informatics and Computing

by

DIMPLE CHEHAL

(Registration No.: 17-YMCA-902001)

Under the Joint Supervision of

Dr. PARUL GUPTA
Supervisor

Dr. PAYAL GULATI
Co-Supervisor

Department of Computer Engineering
J.C. Bose University of Science and Technology, YMCA, Faridabad
Sector-6, Mathura Road, Faridabad-121006, Haryana, INDIA

December 2023

“To my family for their endless
love, support and encouragement”

CANDIDATE'S DECLARATION

I hereby declare that this thesis entitled **A DESIGN OF AN E-COMMERCE RECOMMENDER SYSTEM USING DATA ANALYTICS** *by* **DIMPLE CHEHAL** being submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy in COMPUTER ENGINEERING under Faculty of Informatics and Computing, J.C. Bose of University of Science and Technology, YMCA, Faridabad, during the academic year 2022-2023, is a bona fide record of my original work carried out under guidance and supervision of **Dr. PARUL GUPTA, ASSOCIATE PROFESSOR** and **Dr. PAYAL GULATI, ASSISTANT PROFESSOR**, DEPARTMENT OF COMPUTER ENGINEERING and has not been presented elsewhere.

I further declare that the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

Dimple Chehal
17-YMCA-902001

CERTIFICATE OF THE SUPERVISOR

This is to certify that this Thesis entitled **A DESIGN OF AN E-COMMERCE RECOMMENDER SYSTEM USING DATA ANALYTICS** by **DIMPLE CHEHAL (17-YMCA-902001)** submitted in fulfillment of the requirement for the Degree of Doctor of Philosophy in **COMPUTER ENGINEERING** under Faculty of Informatics and Computing, J.C. Bose University of Science and Technology, YMCA, Faridabad, during the academic year 2022-23, is a bonafide record of work carried out under my guidance and supervision.

I further declare that to the best of my knowledge, the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

Dr. Parul Gupta
ASSOCIATE PROFESSOR
Department of Computer Engineering
Faculty of Informatics and Computing
J.C. Bose University of Science and Technology, YMCA, Faridabad

Dr. Payal Gulati
ASSISTANT PROFESSOR
Department of Computer Engineering
Faculty of Informatics and Computing
J.C. Bose University of Science and Technology, YMCA, Faridabad

Dated:

ACKNOWLEDGEMENT

It gives me immense pleasure to express a deep sense of gratitude towards my PhD supervisor, **Dr. Parul Gupta**, Associate Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana and co-supervisor **Dr. Payal Gulati**, Assistant Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana for their constant guidance and support throughout the PhD learning process. Their vision, meaningful insights, knowledge, experience and timely direction proved beneficial throughout the PhD journey. Their observations and constructive criticism assisted in establishing the overall direction of the research and advancing the investigation in depth. Further, I am grateful for their assiduous efforts in analysing, evaluating and refining the manuscripts.

I would also like to extend gratitude towards **Dr. C.K. Nagpal**, Dean, Faculty of Informatics and Computing, Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, **Dr. Atul Mishra**, Chairman and Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, **Dr. Manjeet Singh**, Chairman and Professor, Department of Computer Application, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, **Dr. Komal Kumar Bhatia**, ex-Chairman and Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, **Dr. Neelam Duhan**, PhD Coordinator, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, **Dr. Ashlesha Gupta** and **Dr. Lalit Mohan Goyal**, PhD Co-coordinator, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana and all other esteemed faculty members of J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana for providing all the necessary facilities, resources and assistance for conducting this research.

I gratefully acknowledge the University Grants Commission (**UGC**), Government of India, for granting research fellowship on time during the period of this work. A special thanks to DRC members for their constructive feedback on the PhD progress reports and the manuscript reviewers for shaping up the ideas with their insightful comments. Also, I am thankful to the members of the IT staff and the department personnel for their prompt assistance and co-operation with the research work.

This endeavor would not have been possible without the unwavering support of my family, friends and colleagues who constantly encouraged me towards the completion of this research work. Heart-felt thanks to my parents, **Mr. Dharampal** and **Mrs. Dhanpati**, for instilling in me a strong desire to approach every challenge in life with a spirit of perseverance and determination; thanks to my brother, **Mr. Sandeep**, sister-in-law, **Mrs. Nisha** for their love, support and invaluable advices, niece, **Priyangna** and nephew, **Priyansh** for keeping me sane throughout the whole process.

Dimple Chehal
17-YMCA-902001

ABSTRACT

Recommender systems are a combination of information retrieval and decision making systems. Just like information retrieval system, these systems return results for a query with the major difference that the query in recommender systems is implicit or not specified by the user explicitly. These systems assist the user to arrive at a purchase decision by suggesting relevant products or services after incorporating their preferences. They considerably assist the users in handling the data overload challenge by limiting the large number of available alternatives for items of interest. In order to gauge the user preferences, implicit sources such as number of product clicks, zooms, bookmarks and explicit sources such as star ratings and product reviews are taken into account. There exists two most widely used categories of recommender systems; collaborative and content based filtering. The former takes into consideration a user's past purchase history and feedback to identify similar users or items and then based on the similar users' or items' inclination for the target user or product generates recommendations. Whereas the latter technique considers a product's characteristics or aspects to identify similar items and generate recommendations for the target user. Star ratings refer to a scale-based (popularly, 1 to 5) feedback mechanism through which users specify their overall experience with a product or service. Systems dependent on star ratings are unable to mine the actual user experience pertaining to products' features or attributes. Unlike star ratings, side information such as product reviews support the aspect level analysis of users' interaction with products. This makes it crucial to build recommender systems that employ the analysis offered by product reviews. As the collaborative filtering systems identify users' or products' neighbourhood, similarity calculation becomes the primary step. The existing similarity measures are based on ratings instead of product reviews. Hence, these measures suffer from the gap of leveraging product reviews to build user or item profile and usage of ratings based similarity measures. There arises a need to formulate similarity measures based on product reviews to maximize the benefit offered by the information contained in them. Recommender systems predominantly face the sparsity challenge due to insufficient data pertaining to user-item interaction. This challenge is overcome by employing the usage of product reviews. However, not all users share product reviews while engaging with the recommender systems and do not share their experience with all the features of the product leading to a novel sparsity problem. This problem due to subjectivity of reviews needs to be addressed in order to improve the performance of underlying algorithms of recommender systems. System users heavily rely on already existing product reviews while making a purchase decision. However, sifting through the large number of reviews is practically impossible. This issue should be addressed by identifying useful reviews so as to decrease the time taken to make a purchase decision. The pandemic led to e-commerce platforms grappling with uncertain client mindsets and minimal interactions. With no existing study on users' emotions and mindset, formulation of government policies and corrective measures were delayed. There aroused a need for understanding the customer's mindset to timely release standard operating procedures for the benefit of all the stakeholders.

The thesis has proposed various methods to solve the problems discussed above. To incorporate review analysis, uninteresting features have been identified by using Latent Dirichlet Allocation topic modeling technique and opinion mining of product (mobile phones) reviews followed by mapping of positive item features, negative item features and features not reviewed with the overall product features. Items with irrelevant features are then not included in the resulting recommendation list. A new similarity measure based on user reviews has been proposed to close the gap occurring due to rating based similarity measures and review based user and item profiles. Aspect based sentiment analysis of product reviews has been used to build user and item profile corresponding to their importance with product's features. The resulting user-aspect and item-aspect matrices' sparsity is mitigated using matrix factorization and auto encoder. The resulting matrices, fed as input to the state-of-the-art collaborative filtering algorithms yield significantly improved results as compared to the original sparse matrices. As there was deficiency of a labelled dataset for performing aspect based sentiment analysis, a manually annotated dataset of mobile phone reviews containing aspect categories and aspect sentiment labels has also been contributed. To identify useful reviews, machine learning algorithms such as Naïve Bayes, Support Vector Machine, Logistic Regression, Random Forest, K Nearest Neighbour, Multi Layer Perceptron and Keras Sequential Model API have been used on derived review features. Also, impact of pandemic on e-commerce platforms has been analysed as it was imperative to understand the users' sentimental and emotional mindset as well as the shift in consumers' purchase behaviour for the benefit of the consumers, the policy makers (stakeholders) and the producers during the pandemic. Thus, a design of an e-commerce recommender system has been proposed that solves the problems of user rating dependent identification of similar users, sparse data, absence of labelled dataset for aspect based sentiment analysis of mobile phone reviews, infeasible browsing of large number of reviews to make purchase decision, absence of attribute level user preference in star ratings and a study of impact of pandemic on e-commerce has been done.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC	Accuracy
ADA	AdaBoost
API	Application Programming Interface
AUC	Area under the Curve
AIU	Aspect Importance for User
AIP	Aspect Importance w.r.t Product
ABCF	Aspect-Based Collaborative Filtering
ABSA	Aspect-Based Sentiment Analysis
AE	Autoencoder
CF	Collaborative Filtering
CBF	Content-Based Filtering
CNN	Convolutional Neural Network
DT	Decision Tree
eWoM	electronic Word of Mouth
ET	Extra Trees
FM	Factorization Machines
GB	Gradient Boost
HTML	Hyper Text Markup Language
KNN	K Nearest Neighbors
LOCF	Last Observation Carried Forward
LDA	Linear Discriminant Analysis
LR	Logistic Regression
MCC	Matthews Correlation Coefficient
MF	Matrix Factorization
MAE	Mean Absolute Error
MSE	Mean Squared Error
MLP	Multi Layer Perceptron
MICE	Multivariate Imputation by Chained Equation
NLP	Natural Language Processing
POS	Part-of-Speech
PCC	Pearson Correlation Coefficient
PMF	Probability Matrix Factorization
PIP	Proximity-Impact-Popularity
RF	Random Forest
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
SemEval	Semantic Evaluation
SVD	Singular Value Decomposition

CHAPTER 1

INTRODUCTION

CHAPTER I

INTRODUCTION

This chapter provides an introduction to recommender system concepts, the motivation behind conducting this research, the problem statement and the techniques employed to complete this research.

1.1 GENERAL

In today's digital time, e-commerce users are overloaded with information about products information. As the items, their descriptions, and user feedbacks is present in large quantities, the size of the information becomes humongous. User feedbacks refers to star ratings and product reviews given by the users. Item or product ratings represent a user's satisfaction level on a scale, which usually ranges from 1 to 5, with 1 star being the lowest and 5 stars being the highest satisfaction level. First-hand experience of a user's interaction with a product is represented in the form of textual reviews. Both type of feedback represents satisfaction and experience of customer with a product. Today, to-be consumers depend on previous reviews for their purchase decision. Recommender systems are of great help to downsize the size of information as they suggest only personalized and relevant alternatives to the users according to their preferences (Jalili, Ahmadian, Izadi, Moradi, & Salehi, 2018; Portugal, Alencar, & Cowan, 2018; Batmaz, Yurekli, Bilge, & Kaleli, 2019).Such systems offer personalized choices for customers after taking into account their interests and disinterests (Ricci et al., 2011). This user interest towards a new product is captured through the users past purchase history and feedback of purchased products. The filtering of alternatives so as to suggest relevant and personalized products distinguishes the different types of recommender system. Based on different types of filtering techniques, recommender system majorly consists of three filtering methods namely Collaborative Filtering (CF), Content-Based Filtering (CBF) and hybrid recommender system (Cacheda, Carneiro, Fernández, & Formoso, 2011; Sohail, Siddiqui, & Ali, 2017; da Silva, de Moura Junior, & Caloba, 2018; Kumar, Kumar, & Thakur, 2019). Combining the benefits of CF and CBF techniques leads to the hybrid filtering technique. Types of recommender system are shown in Figure 1.1 (Taghavi, Bentahar, Bakhtiyari, & Hanachi, 2018; Sinha & Dhanalakshmi, 2022; Papadakis, Papagrigoriou, Panagiotakis, Kosmas, & Fragopoulou, 2022).

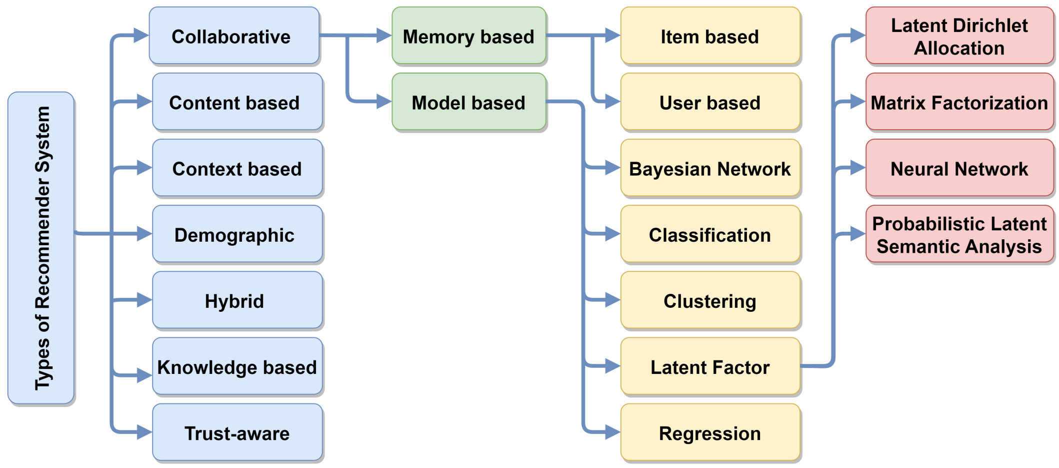

Figure 1.1: Recommender System Types

CF techniques take into consideration similarities between users or items while CBF techniques need domain knowledge to help generate recommendations. CF-based recommender system assumes that if users have similar preference in the past they will follow the same pattern in the future as well. CF techniques are further classified into model-based and memory-based methods. Model-based method creates a model to discover underlying hidden patterns that explain the ratings of items by users. It involves building of machine learning or data mining model such as Bayesian, clustering, Singular Value Decomposition models and Decision Trees for predicting the user's preferences. Memory-based methods are traditional methods that use similarity mechanism to compute item or user similarity. It takes into consideration the entire user-item ratings to identify similar users/items. Post identification of similar entities, their past ratings is used to generate recommendations. CF engines do not require domain expertise and have the ability to generate cross-domain recommendations. Also, such engines need only user behavior information to generate recommendations. CBF models are dependent on items' content and independent of user specific information. This independence makes CBF more scalable as compared to CF systems. Also, CBF has the ability to recommend items that have received less interest from other users, thereby capturing the interest of specific user.

However, CF techniques are unable to deal with data sparsity, scalability and cold start problem (Nehete & Devane, 2018). CBF is unable to deal with limited content analysis as information required to build user profile is not easily available and tends to recommend items which are very similar to the items liked previously by the user(s). As the items are compared for similarity based on their features described textually, this mandates the text be present in easily parse-able format or the features of an item should be associated manually which is practically infeasible given the size of items. Another problem faced by CBF is dimension reduction due to large number of users, items and ratings in recommender system (Bunnell, Osei-Bryson, & Yoon, 2019; Shen, Zhang, Yu, & Min, 2019). Also, CBF is unable to handle the problems of overspecialization and cold-start problem (da Silva et al., 2018; Kumar et al., 2019; Duan, Jiang, & Jain, 2022). 88 In addition to the three main categories of recommender systems, there also exist demographic, knowledge based, context based and trust-aware recommender systems. In the demographic recommender system, the user classification is based on their personal information to generate recommendations. In the knowledge based recommender system, the suggestions generated utilize specialised domain knowledge about users' requirements, products' features and how features fit into the users' preferences. In the context based recommender system, the focus is on contextual data gathered from sensors, position, time, location of the user, etc., to generate recommendations. In the trust-aware recommender system, social networks play a key role in providing suggestions by taking into account the preferences of the target user's friends (Sohail et al., 2017).

The problem of sparsity arises when sufficient user-item interaction information is not available as compared to the number of users and items that constitute the system. In other words, very less number of ratings by users for items than the number of available users and items generates a sparse user-item rating matrix. This insufficient interaction information creates problem in similar user computation followed by generation of recommendations. Cold-start problem occurs when new users or items are added to the system. As the preference of these newly added users is not available initially, similar user computation is difficult. Cold-start items are not recommended until their ratings are available, thereby possibly devoiding users of good recommendations (Sachan & Richhariya, 2013; Da'u & Salim, 2020). In order to overcome the sparsity problem, recommender system should ensure rating of all items by its users. However, not all users understand the importance of explicit feedback or confirm to the feedback cycle. This necessitates developing methods to mitigate the sparsity problem.

Recommender systems based on user reviews are known to alleviate the sparsity and cold-start problem (Chen, Chen, & Wang, 2015). This study makes use of product reviews to meet the research objectives. The user or product profile built through the help of user reviews are used to augment the available product ratings. Also, with the help of user reviews, interest of the user towards a product or service can be gauged in terms of the product's or service's features. This feature related information is not available in recommender system based on product ratings. Another problem prevalent in recommender system is infeasible browsing of all product reviews due to information overload. In the existing recommender system, votes gained by a review can be used to browse product reviews. But, due to factors such as humongous volume of electronic word of mouth, voluntary helpfulness voting mechanism, level of visibility and review recentness, all reviews do not receive this vote. Categorizing of product review according to their usefulness will helps the user to browse the relevant user reviews. In review-based recommender system, the reviews can be leveraged to identify similar users based on the reviews provided by them, instead of the star ratings.

While employing reviews to build a recommender system, a new kind of sparsity problem surfaces with the usage of product reviews. This new problem originates due to subjectivity of reviews, that is, few product features reviewed by the users in their feedbacks. That is, the information contained in reviews is sparse as all the product aspects are not reviewed by the users. Solving the sparsity problem arising due to subjectivity helps to improve the performance of recommendation system.

Apart from the above well researched problems, there is another uncalled problem that all the platforms face together, that is, pandemic such as Covid-19. During such pandemics, user interaction is limited due to imposed lockdowns. In such scenarios, it becomes essential to record the users' feedback sentiment from time to time so that the concerned government authorities can take necessary actions. These actions or events impact the emotional well being of a person. Through this study, the authorities after knowing citizen's emotional state can chalk out policies beneficial for the users. Also, e-commerce stakeholders can adjust according to their region and regulate products demand and supply. As the user feedback on e-commerce platform is limited, tweets from social media platform, Twitter, can be utilised to analyse users' mindset through topic modelling techniques such as Latent Dirichlet Allocation (LDA).

To establish the platform and e-commerce domain of study, a short survey has been conducted. The survey has been responded by 108 respondents. Out of 108 respondents, only 1 respondent states not using e-commerce platforms for online shopping. The respondents state Amazon to be the most used platform for their online shopping as compared to other Indian e-commerce platforms such as Flipkart, Myntra, Ajio, TataCliq and Nykaa. 75.2 % respondents prefer buying mobile phones through e-commerce platforms. Out of all the verticals such as mobile phones, laptops, books, apparel, cosmetics/beauty, security appliances, home appliances, grocery, health, sports/fitness equipments, toys/baby products and pet products, the majority of the respondents, 33.3% prefer buying mobile phones from e-commerce platform as compared to other domains. This helps to identify Amazon and mobile phones as platform and e-commerce domain for this study. While over 75% of the respondents are satisfied with the recommendations being suggested to them, around 25% of the respondents indicate neutral response towards the recommendations displayed to them, highlighting the improvement that should be done in the recommendation process. The questionnaire of the short survey is added in the Appendix A for reference.

1.2 MOTIVATION

As seen above, there are several challenges that limit the performance of recommender systems. The motivation behind the research work is as follows:

1. **User rating dependent identification of similar users**
   Selection of a similarity measure in collaborative filtering is an important task (Fkih, 2022). The efficiency of a recommender system depends on the similarity measure chosen. Rating-based recommender systems consider similarity measures based on commonly rated items by its users. Reviews-based recommender system has established its supremacy over rating-based recommender system (Chen et al., 2015). In the existing literature, recommender systems that employ usage of reviews for user similarity comparison are deficient. A gap is created due to usage of reviews as side information for building user/item profiles and user similarity computation using the provided ratings. Hence, there is a need to define similarity measures that rely on user reviews.

2. **Sparse rating data**

   Recommender systems employ user feedback to build on user and item profile. But, such systems grapple with sparse data as very few ratings are provided by users as compared to the quantity of items that exist on the platform that makes use of such recommender systems. In order to deal with this sparsity, user reviews are used as side information. This helps in augmenting the user/ item profile and improves the recommendation process. However, just like ratings, the reviews provided by users are also less in quantity as compared to the items present in the system. Further, the reviews are incomplete as not all aspects/ features of a product are reviewed by the user. This leads to a novel sparsity problem due to subjectivity of reviews. This problem needs to be addressed in order to improve the performance of collaborative filtering algorithms thereby improving the overall recommender system (Idrissi & Zellou, 2020; Singh, 2020; Yang, Zhou, & Cao, 2020).

3. **Absence of labeled dataset for Aspect-Based Sentiment Analysis of mobile phone reviews**

   Aspect-Based Sentiment Analysis (ABSA) helps in identifying the contributing aspect(s) and their corresponding polarity, thereby providing a more detailed analysis of customer's inclination towards feature(s) of a product. In the literature, ABSA has been performed on movie reviews, digital cameras, restaurants, telecom, consumer electronics and museum (Pontiki et al., 2015) on various languages such as Czech, Bangla (Rahman & Dey, 2018), French and Hindi (Akhtar, Ekbal, & Bhattacharyya, 2018). But, no annotated or labeled dataset, indicating the aspects and sentiments in a review, specifically for mobile phone domain in English language was encountered, the domain identified by the authors for their contribution. The growth of labeled datasets was observed to be less owning to the human involvement. A need for labeled dataset for performing aspect level sentiment analysis of mobile phone reviews given by customers in English language was identified. The contribution of such an annotated dataset would also help ML algorithms as part of supervised learning to predict the output, given the input text to be classified and the output label or aspect categories/sentiments.

4. **Infeasible browsing of large number of user reviews to make purchase decision**

   Online user reviews are known to affect the purchase decision of users as established by the researchers' community. Due to large number of reviews posted it is not possible for the users' to go through all the reviews. The users can make better purchase decisions by browsing useful reviews alone (Mitra & Jenamani, 2021). Determination of useful reviews is an open problem, dependent on the voting mechanism as not all reviews receive a vote from the users and new reviews are not voted as soon as they are posted. Hence, a solution to this problem is required for the benefit of the users as well as the stakeholders involved.

5. **Absence of attribute level user preference in star ratings**

   A large number of recommender systems employ star ratings to gauge the inclination of users towards products/services. However, the effectiveness of ratings becomes limited in the sparsity scenario and due to their inability to capture feature/aspect/attribute level preference of users towards a product/service. As a result items are recommended to users based on the overall preference ignoring the underlying details. Hence, a review analysis based mechanism is needed to capture the feature driven preference of users as this information exists in reviews instead of ratings (Chen et al., 2015).

6. **No existing study of impact of pandemic on e-commerce**

   During the pandemic and subsequent lockdowns, several sectors are financially impacted. As the lockdowns are imposed, it becomes impossible to understand the clients' mindset due to absence of any sort of user interaction. The stakeholders involved require users' response to formulate policies and take corresponding action and corrective measures. User response analysis is unavailable as the pandemic emerges. Hence, it becomes imperative to understand the users' sentimental and emotional mindset as well as the shift in consumers' purchase behaviour for the benefit of the consumers, the policy makers (stakeholders) and the producers.

1.3 PROBLEM STATEMENT

The importance of recommender system is well recognized in industry as well as academia. Incorporating the usage of such systems benefits the customers and the business involved. They help to identify and convert potential customers into actual customers by capturing their preferences for items of interest thereby influencing the purchase decision. Such systems are being used in various sectors such as e-commerce, health, education, entertainment, tourism, food etc. However, the systems still grapple with challenges such as identification of similar users through star ratings, sparse data availability, absence of annotated datasets for review analysis, browsing of large number of user reviews to arrive at a purchase decision, absence of attribute level preference information in star ratings deficiency of studies of impact of a pandemic. As e-commerce platforms are majorly used to purchase mobile phones and a sizeable chunk of users indicate the need for improvement in the suggested recommendations, there arises a need to design a recommender system for e-commerce through data analysis techniques that resolves the identified shortcomings.

1.4 SYSTEM CONFIGURATION

For carrying out this research, the following hardware configuration and softwares were used: **Hardware configuration:** Windows 10, Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz Processor, 64-bit operating system, x64-based processor **Software:** Python 3.7.1, Jupyter, R, Tensorflow, Keras, Scikit-learn, Numpy, Pandas

1.5 THESIS ORGANIZATION

The thesis consists of nine chapters as shown below:

1. **Chapter 1** discusses introductory concepts about recommender system, outlines the related challenges, motivation behind this research, problem statement, system configuration and organization of this thesis.

2. **Chapter 2** presents the literature studied for achieving the objectives. The chapter begins with explaining the various types of recommender system that exist in literature. This is followed by the comparison of types of recommender system with respect to the challenges faced by them. Further, the pros and cons of existing similarity measures are provided. Previous studies on sparsity removal techniques and identification of useful reviews are also discussed in the chapter.

3. **Chapter 3** lists and describes the objectives of this research.

4. **Chapter 4** describes the proposed approach based on similar user identification using product reviews. As part of the approach, sentiment score of the product review is computed and used to identify similar users followed by prediction of sentiment score for items not reviewed by the users.

5. **Chapter 5** focuses on the proposed approach to handle sparsity problem in review-based recommender system leading to improvement in the performance of the base algorithms used in recommender system. Aspect-based sentiment analysis of user reviews is utilised to compute the inclination of users and items towards product aspects. The resulting user-aspect and product-aspect preference matrix is matrix factorized and auto encoded to alleviate the sparsity problem. The non-sparse matrices significantly improve the performance of state-of-the-art recommender system algorithms.

6. **Chapter 6** deals with identification of useful product reviews through machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, AdaBoost, Gradient Boost, Extra Trees, K Nearest Neighbor and Linear Discriminant Analysis. Pre-processing, feature engineering, model training, model testing on the original and derived features is performed and then the models are evaluated for accuracy, area under the curve, precision, recall, f1 score, Kappa score and Mathew correlation coefficient metrics

7. **Chapter 7** presents a proposed method to incorporate analysis offered by product reviews in recommender system. Opinion mining of product reviews

using vadersentiment python library and topic modeling of product reviews using Latent Dirichlet Allocation (LDA) is done to identify uninteresting product features. Such features are not to be included in the final recommendation list.

8. **Chapter 8** covers the study conducted to analyse the impact of a pandemic on e-commerce platform. Such a study helps to understand users' mindset in the middle of a pandemic wherein the functioning of such platforms is minimal to contain the outbreak's spread. As the users' interaction with the e-commerce platforms through product reviews is negligible, their perspective for e-commerce is mined through social media tweets.

9. **Chapter 9** summarizes the contribution of this research and suggests extension of this work that can be carried out in the future.

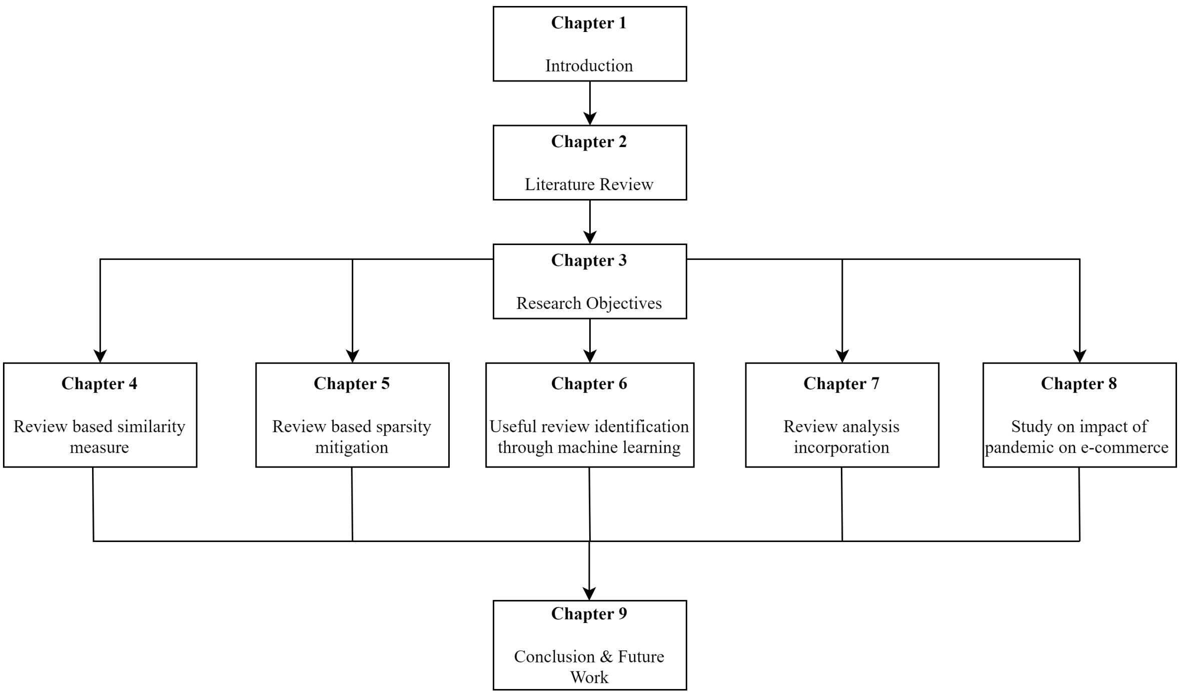

Figure 1.2: Organization of Thesis Chapter-Wise

The thesis consists of nine chapters as shown in Figure 1.2. Lastly, bibliography, brief profile of the research scholar and list of publications as part of this research are included.

CHAPTER 2

LITERATURE SURVEY

CHAPTER II

LITERATURE SURVEY

This chapter presents the studied literature. It begins with explaining the various types of existing recommender system. This is followed by the comparison of recommender system with respect to the challenges faced by them. User or item similarity identification is a major step in recommendation generation process. Advantages and disadvantages of existing similarity measures have been provided. Importance of incorporation of review analysis, previous studies on sparsity removal techniques and identification of useful reviews are also discussed in the chapter.

2.1 RECOMMENDER SYSTEM

Nowadays, there are a large number of online platforms offering a wide variety of information in the form of products or services to the users. The process of finding relevant and helpful products or services has become difficult due to the presence of abundant data (Roy & Dutta, 2022). This data overload makes it difficult for the information system to work efficiently without the usage of recommender systems. Previously, recommender systems were studied as a subfield of data mining and information filtering. It wasn't until the 1990s that it was officially acknowledged as a legitimate separate field of study. Recommender system has become a topic of interest in both the industry as well as the academia. The widespread use of e-commerce programmes like Netflix, Pandora, and Spotify, YouTube for entertainment and Amazon for purchase recommendations has made recommender system well-known among the general public at large (Bunnell et al., 2019). Originally, the purpose of recommender system was to help reduce cognitive overload by retrieving only the most relevant and valuable items from a vast array of options. In general, recommender system serves as information-filtering tools, providing users with relevant and tailored content or data. The primary objective of recommender system is to reduce the time and effort required by users to find relevant content on the internet.

In the era of ”big data,” the volume of information available to consumers makes it increasingly difficult for Information System (IS) users to comprehend and evaluate all of the options. In this context, recommender systems offer a realistic technique of:

1. Filtering through vast quantities of data
2. Determining user preferences
3. Suggesting users with relevant, useful, and tailored recommendations

Few of the commercial platforms that employ usage of recommender system are shown in Figure 2.1.

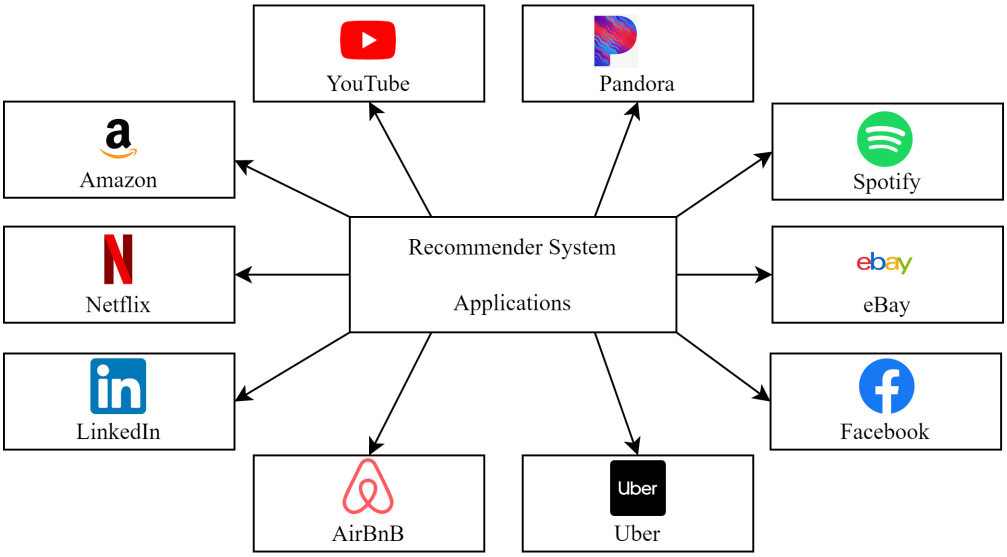

Figure 2.1: Commercial Platforms that Utilize Recommender System

Recommender system is a specialised sort of Artificial Intelligence (AI); Decision Support System (DSS) that can assist users with insufficient personal experience or knowledge in making judgements by processing vast volumes of data to aid in their decision-making processes. The requirements of offering customized recommendations that are interesting, relevant, and useful to the user, rated according to the user's preference, distinguish recommender system from simple Information Retrieval (IR) techniques (del Carmen Rodríguez-Hernández & Ilarri, 2021). Recommender system majorly involves two components namely, users and items, wherein a user assigns a rating or preference to an item. These systems facilitate the control of information overload by autonomously obtaining and proactively adjusting information to user preferences (Adomavicius & Tuzhilin, 2005). This is achieved by employing one or more filtering techniques, such as Collaborative Filtering (CF), Content-Based Filtering (CBF), and hybrid filtering (Jalili et al., 2018).

Recommender system is essentially a part of information retrieval, information system, as shown in Figure 2.2.

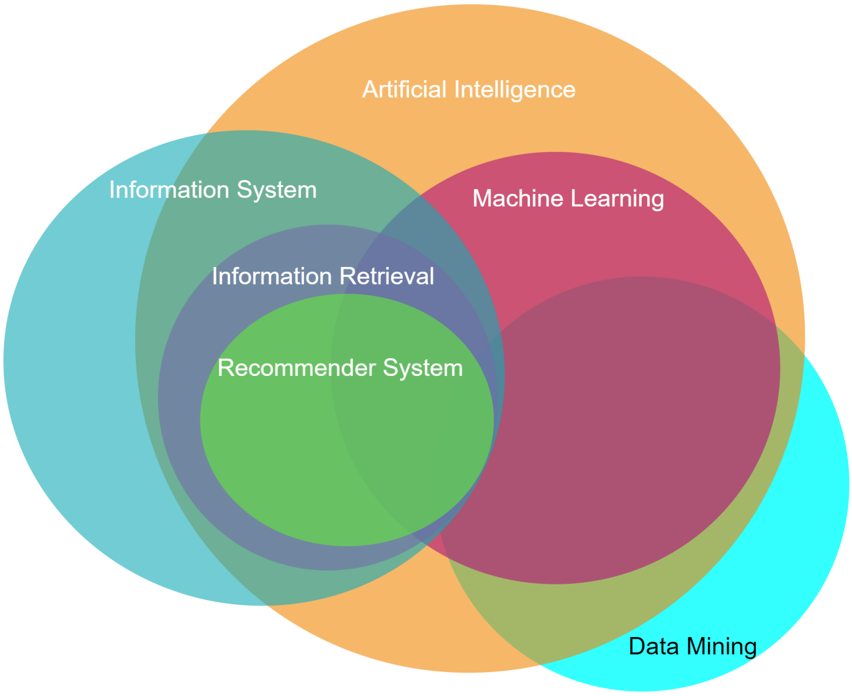

Figure 2.2: Fields Related to Recommender System

The various stages that are involved in the process of CF and CBF, are shown in Figure 2.3:

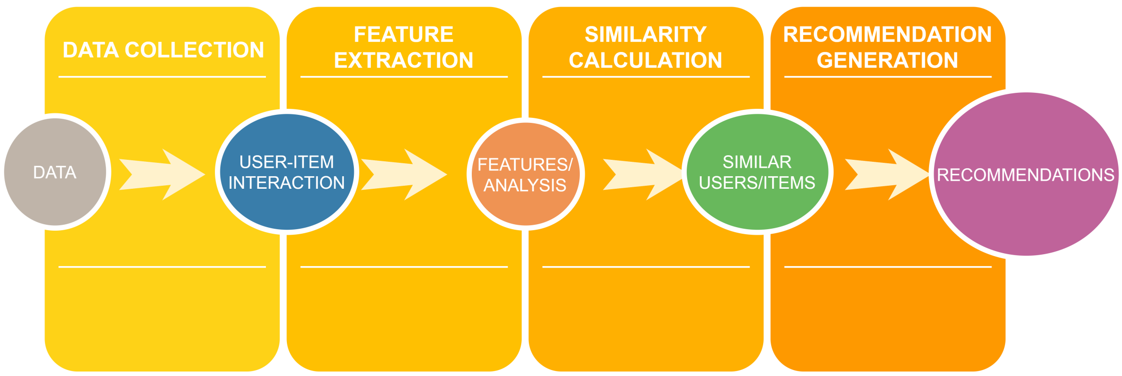

Figure 2.3: Major Stages Involved in Recommender System

1. **Data acquisition**:
   CF and CBF require a dataset of customer-product interaction in order to generate suggestions (such as, ratings, clicks, views, zooms and reviews)

2. **Feature extraction**:
   In CBF, the system examines the features of the objects (for example, battery,

screen resolution, camera and price) in order to recognise trends and create recommendations. In the process of CF, the system examines the interactions between users and items in order to recognise trends and produce recommendations.

3. **Calculation of similarity**:
   In CBF, item features are utilised to determine the items that are similar to the target item. Unseen item's features are compared for similarity with the target item's features. In CF, users similar to the target user are determined by considering similar calculating the degree to which the target user's interactions are similar to that of other users.

4. **Generation of Recommendations**:
   Both CBF and CF, depending on the similarity calculations, provide a list of things that the user is suggested to view based on the list.

Product ratings, item characteristics and content, user registration information, and information about social ties are some examples of the numerous kinds of data that are recorded in recommender system. In case such data is absent, data is gathered through the following sources (Taghavi et al., 2018):

1. **Explicit sources**:
   The actual data about the system's items or users (such as item features, demographic information of user), or an explicit user experience in the form of feedback, such as ratings and reviews for purchased items.

2. **Implicit sources**:
   The behavioural usage data of the user, such as his or her past purchasing behaviour, the location of the user's browsing session, user's click-zoom ratio, or the detection of the user's opinion about a product without the user sharing this information explicitly himself.

Mostly, either implicit or explicit approaches are utilised in order to obtain ratings from users. The user's interaction with the items is used to infer the user's implicit ratings, which are collected in an indirect manner from the user. On the other hand, explicit ratings refer to straightforwardly provided preference on a finite scale of rating values. Usually, platforms derive implicit evaluations for items through click-stream data or from the bookmarking of a web page and so on. The vast majority of recommender system utilise both explicit and implicit means to collect ratings from users. The utility matrix is a user-item matrix that contains the user feedbacks or ratings that were supplied by the user (Hernández-Rubio, Cantador, & Bellogín, 2019).

The technique of analysing the preferences of similar users, in order to generate recommendations is known as CF. The same has been represented diagramatically in Figure 2.4:

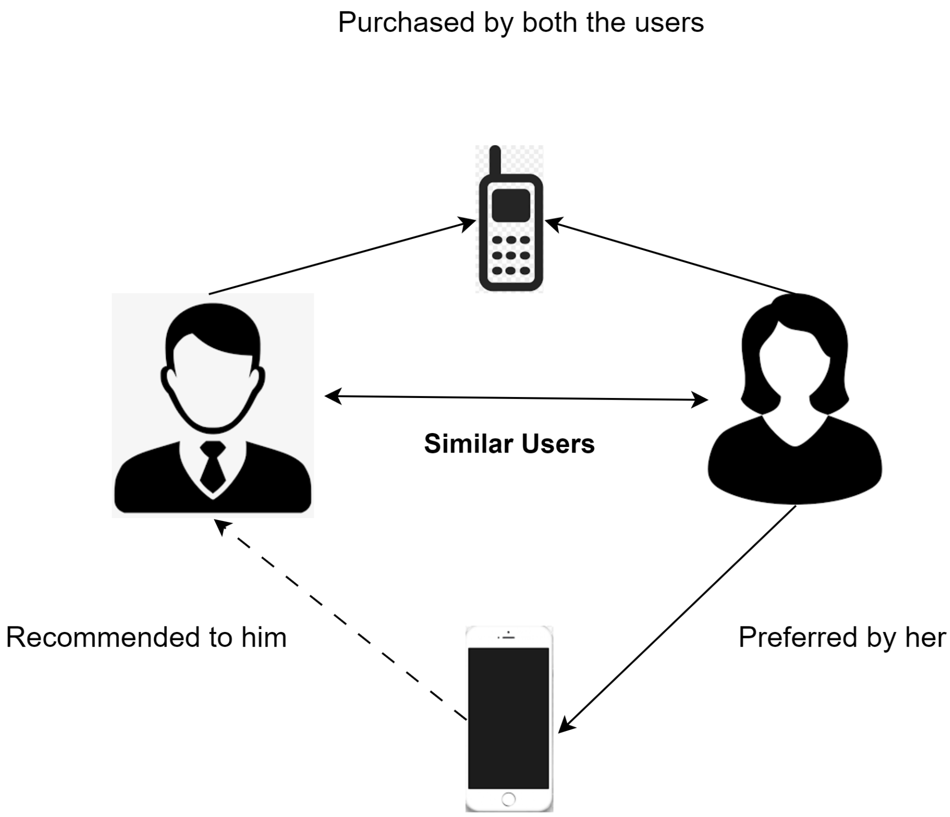

Figure 2.4: Functioning of Collaborative Filtering Technique

Recommendations can be generated by CF methods by comparing the profile of a user to the profiles of other users and comparing them on relevant interests and preferences. This is often accomplished through the utilisation of similarity metrics. CF-based traditional systems suffer from the cold start problem. Additionally, they suffer from privacy concerns due to the fact that user data has to be shared in order to generate recommendations (Bunnell et al., 2019). By utilising dimensionality reduction techniques and model learning approaches, model-based strategies are able to solve some of the more classic issues that are associated with recommender system (Nilashi, Ibrahim, & Bagherifard, 2018). These issues include sparsity and scalability (Idrissi & Zellou, 2020). The scalability of matrix factorization techniques is beneficial to CF systems since it helps to lower the computational complexity of the process of generating recommendations. Additionally, there are a variety of methods that can be utilised to assist scale up CF systems and make them more efficient. Some examples of these methods include parallelization and distributed computing.

CBF suggests similar items to the user based on the characteristics of the recommended items. CBF approaches employ knowledge engineering techniques for the extraction of features, tags, or meta-data information from products of a recommender system database and then compare those items with users based on their past ratings of other items, , as shown in Figure 2.5. Both of these approaches are susceptible to a number of intrinsic difficulties and obstacles. In addition, this method is limited in its ability to increase the consumers' existing preferences or interests (Roy & Dutta, 2022). Since a user's profile is unique to that user, this algorithm does not require the profile information of other users, as they have no bearing on the suggestion process. This maintains the confidentiality and security of user data.

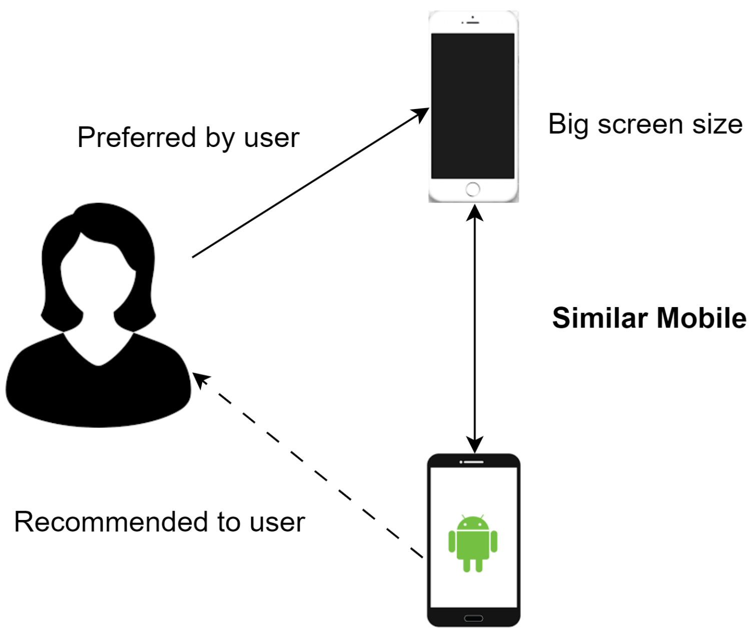

Figure 2.5: Functioning of Content-Based Filtering Technique

2.2 TYPES OF RECOMMENDER SYSTEM

In order to address issues with recommender system such as cold-start and ratings sparsity, a variety of novel methods for recommender system have been developed. These include demographic, knowledge-based, context-aware and trust-based recommender system (Sohail et al., 2017):

1. **Demographic**:
   Such a recommender system classifies users according to the personal characteristics they have provided in order to make recommendations based on their

demographic profiles. The underlying idea here is that various demographic subgroups ought to be targeted with offers that are tailored to them specifically. This form of recommender system functions in a manner that is somewhat comparable to that of content-based systems; however, the advantage of this approach is that, unlike collaborative and content-based methods, it does not necessarily require user ratings histories (Taghavi et al., 2018). Such systems face the stereotyping challenge. In the context of recommender system, the risk of ”stereotyping” refers to the possibility that demographic filtering may recommend items or content based solely on a user’s demographic group (such as age, gender, or location), without taking into account the user’s individual preferences or behaviour. For example, a younger user may be more likely interested in mobile phones with mobile phones with high end camera quality than older users. For instance, recommender system recommending a specific mobile phone brand to a user based solely on their age or gender, without taking into account the person’s specific tastes might potentially lead to stereotyping and limiting the variety of choices available. It is possible that as a result of this, the system will not be able to accurately capture the specific preferences of the user, which may result in the user being dissatisfied with the recommendations (Bunnell et al., 2019).

2. **Knowledge-based**:
   This recommender system places an emphasis on information sources that are not included in CBF and CF methods. This is done by generating suggestions based on specialised domain knowledge about the requirements of the users, the features of the objects, and how these features can fit the requirements and preferences of the users. As they are not dependent on user ratings, this technique has a tendency to perform better than the rest. On the other hand, in order to maintain this edge, they need to be associated with learning components that employ usage of the human-computer interaction log (Sohail et al., 2017). This technique relies on the explicit information of the user’s preferences and behaviour. However, like other recommender system techniques it also faces challenges, primarily the knowledge acquisition problem, that is, to obtain correct and comprehensive information about the user of the system.

3. **Context-based**:
   This recommender system focuses on supplementary pieces of contextual data such as time, position, wireless sensors etc. The contextual data could be gathered through the use of techniques such as data mining, explicit and implicit feedback, or both. Mobile applications, for instance, make extensive use of geo-location information in order to produce recommendations by taking into

consideration the location of the user.

4. **Trust-aware**:
   This recommender system takes into account the preferences of target user's friends. It is known that people are more likely to take the recommendations made by their friends than those made by others who share similar characteristics but remain nameless. The popularity of social networks continues to rise on a daily basis, which has led to an increased interest in community-based recommender system, also known as social recommender system.

5. **Hybrid**:
   This recommender system combines the approaches described above in order to obtain improved levels of performance. A hybrid recommender system combines two different methods, with the goal of utilising the positive aspects of one method to compensate for the negative aspects of the other method. For instance, CF is unable to manage new items that do not yet have any ratings, whereas the CBF does not experience any such difficulty because the recommendations are based on the readily accessible item characteristics.

2.3 COMPARISON OF TYPES OF RECOMMENDER SYSTEM

Recommender systems face several challenges such as cold-start, grey sheep users, over-specialization, privacy concerns, scalability, shilling attack, data sparsity, synonymy, transparency etc.

1. **Cold-start**:
   The non-availability of user preference in the form of ratings, reviews or other implicit sources of a new user or a new item is referred to as the cold-start problem (Hernández-Rubio et al., 2019; Mewada & Dewang, 2021). CF-based recommender system employs past purchase history of a user. Due to this dependency, recommendations for a new user is not generated as no previous information for that user is available. CBF doesn't face the cold-start problem for a new item as it considers the features or characteristics of an item rather than the explicit user provided ratings.

2. **Grey Sheep**:
   The user(s) whose preferences is/are significantly different from the rest of the users of a system is/are known as grey sheep user(s) (Tahmasebi, Ghazvini, & Esmaeili, 2018). This makes it challenging for the recommender system to

predict their preferences or generate recommendations pertinent to the user's interest. Due to the dependence on recognising patterns and similarities in user behavior in order to provide suggestions, the CF approach is the one that most frequently faces this problem. As other recommender system such as CBF, demographic filtering, knowledge-based filtering, context-based filtering, and trust-aware filtering do not rely on recognising patterns of user activity throughout the system; they are less likely to be affected by this problem (Alabdulrahman & Viktor, 2021).

3. **Over-specialization**:
Unlike CF, CBF systems suffer from the over-specialization phenomena as they have a propensity to only suggest items that are similar to those that have already been rated by the user. Introduction of a new item to the previous preferred item pool is considered to be an easy solution to this problem (Bunnell et al., 2019). Context-aware and Trust-based recommender systems are considered to overcome this problem of over-specialization.

4. **Privacy**:
Recommender system relies heavily on users' preference and related information such as star ratings, reviews, implicit feedbacks, social connections and demographic data. Concerns about users' privacy can arise from the collection of such kind of data as the users may not feel at ease at sharing of this information. Knowledge-based and Content-based recommender systems do not face this issue as there is no employment of user specific data to generate recommendations.

5. **Scalability**:
A system which is able to accomodate increasing number of users and items, is said to be scalable. CF and CBF face scalability issues when working on huge amount of data (Singh, 2020). When generating suggestions, CF depends on studying the activity of all users in the system. This analysis is computationally expensive and time-consuming due to increased number of users and items in the system. In a similar manner, CBF relies on examining the characteristics of each and every item in the system, which may likewise become computationally expensive and sluggish as the number of items in the system continues to expand. Filtering techniques such as demographic filtering, knowledge-based filtering, and context-based filtering are generally less prone to scalability issues than other filtering techniques. This is because these techniques frequently make use of simpler rules or heuristics to generate recommendations based on user or item characteristics. Trust-based systems can

also become computationally expensive and slow as the number of users and items in the system grows. This is because they may require analysing trust relationships among a large number of users, which can take a lot of time. On the other hand, the unique scalability features of a recommender system might be contingent on the particular design and implementation of that system. Dimensionality reduction and clustering-based techniques to discover people in tiny clusters rather than in the entire database are two typical methods that are used to tackle the scalability problem (Nilashi et al., 2018).

6. **Shilling Attack**:
   A shilling attack is said to have occurred when users offer a lot of positive recommendations for their own products or services and provide a lot of negative recommendations for their competitors' items. It would be beneficial for CF systems to have safeguards that deter the occurrence of phenomena of this nature. This issue emerges when a dishonest user pretends to be someone else in order to get access to the system and then rates items dishonestly (Sundar, Li, Zou, Gao, & Russomanno, 2020). Trust-based systems are susceptible to assaults by malevolent users, who may attempt to control the system by forming fake trust connections or providing misleading trust ratings. These attacks can make the system open to manipulation.

7. **Rating Sparsity**:
   Traditional CF systems are based on explicit feedback such as user ratings. Product ratings are a reflection of a user's experience with an item. In case of large number of users and items in a system, it is challenging to collect sufficient information about each user's interaction with each item, leading to a deficient or sparse user-item rating matrix (Cacheda et al., 2011; Singh, 2020), . As CF requires a considerable quantity of data on user-item interactions in order to create reliable suggestions, it suffers from data sparsity challenge. This results in a lack of information about users or items, making it challenging for CF to effectively forecast the preferences of such individuals or offer appropriate suggestions. Whereas, content-based filtering, demographic filtering, knowledge-based filtering, context-based filtering, and trust-aware filtering are generally less susceptible to sparsity problems than other types of filtering as they make recommendations based on user or item features or other external information that is not dependent on user-item interactions. Knowledge-based systems base their suggestions on domain specific information which can be limited or incomplete. Moreover, it is also difficult to gather information about the trustworthiness of users, which is required by trust-based systems. This is especially the case in networks that are large and diverse. Further, it is possi-

ble that there is a lack of contextual data, which might make it challenging to effectively model user preferences and to provide reliable suggestions. Hence, there is a possibility that apart from CF technique, remaining techniques might also experience sparsity issues if there is insufficient information available for particular user or object characteristics. A number of different approaches, such as demographic filtering, singular value decomposition, and the use of model-based collaborative procedures, are all viable options for resolving this issue (Idrissi & Zellou, 2020; Yang et al., 2020).

8. **Synonymy**:
   This issue arises when identical or comparable objects are referred to by varying names. CBF is the filtering approach most frequently affected by the synonymy problem, as it analyses the content or characteristics of objects to produce suggestions. In this method, products are suggested based on their resemblance to other items in which the user has already expressed interest. However, if objects with comparable content or qualities are described using different terms or synonyms, the algorithm may be unable to effectively recognise these similarities, resulting in a less effective suggestion. CF is similarly affected by this problem, which diminishes its performance. The SVD approach, namely the Latent Semantic Indexing (LSI) method, is able to address synonymy issues (Sharma, Kumar, & Chand, 2017). In general, demographic filtering, knowledge-based filtering, context-based filtering, and trust-aware filtering are less prone to the synonymy problem than CBF, as they do not rely heavily on examining the content or characteristics of items.

9. **Transparency**:
   The term ”transparency problem” refers to the problem of describing or analysing the rationale behind a recommendation made by a recommender system. The filtering methods that most commonly face this issue are CF and CBF (Cheng, Chang, Zhu, Kanjirathinkal, & Kankanhalli, 2019). It is often difficult to understand the specific reasons behind a particular recommendation when CF is used as it relies on complex algorithms and models to identify patterns and similarities in user behaviours. It is often quite easy to explain to consumers the reason of recommendation of a certain item when using CBF as the item features or characteristics are simple and straightforward to comprehend for the users. For instance, one straight-forward method is to inform customers about the item features that can pique their interest in the item that has been recommended to them. The collection of content information across a variety of application areas, however, can be time demanding. CBF may also utilise complicated models to assess the content or characteristics of

things and generate suggestions based on a mix of criteria that are difficult to comprehend. The demographic filter, the knowledge-based filter, the context-based filter, and the trust-aware filter are generally more transparent than other types of filters because they rely on particular user or item characteristics or other external information that can be more easily explained or interpreted. However, the precise traits of openness and transparency that a recommender system possesses might be contingent on the particular design and execution of that system.

The major challenges faced by the different recommendation techniques are listed in Table 2.1:

Table 2.1: Comparison of Recommender System Based on Existing Challenges

Issues	**CF**	**CBF**	**Demographic**	**Knowledge-based**	**Context-based**	**Trust-aware**
Cold-start	Yes (User & Item)	Yes (User)	No	No	No	No
Grey Sheep	No	No	No	No	No	No
Over-specialization	No	Yes	Yes	No	No	No
Privacy	Yes	No	Yes	No	Yes	Yes
Rating Sparsity	Yes	No	No	No	No	No
Scalability	Yes	Yes	No	No	Yes	Yes
Synonymy	Yes	Yes	No	No	No	No
Transparency	Yes	No	No	No	No	No

Even while recommender system is gaining a lot of interest in commercial and real-life applications, further research and development is still required for these systems to be able to be used effectively in complicated contexts, and a lot of commonly occurring problems such as rating dependency, efficient user similarity measure, data sparsity and review usefulness still need to be solved. It is possible that the lack of literature concerning the recommender system problems categorization is slowing down the progression of research in this growing area of study.

Despite all of the research that has been conducted within the field, recommender system is still plagued by a number of problems that continue to present formidable obstacles in the way of striking the ideal balance between user acceptance and the business performance objectives of the system owner.

2.4 SIMILARITY MEASURES

CF recommender systems take into consideration the user's purchase history as well as the user-item rating matrix when computing a user's preferences. This allows for a more accurate representation of the user's tastes. The process of identifying users or things that are similar to one another is an important stage in this filtering strategy. As a result, the utilisation of a similarity measure has to be acceptable so that it can supplement the area of application.

(Liu, Mehandjiev, & Xu, 2011; Alqadah, Reddy, Hu, & Alqadah, 2015) used cluster based approaches to calculate user similarity based on provided ratings. (Wang, Deng, Gao, & Zhang, 2017) stated the techniques to identify similar users based on user-item rating matrix as Cosine Similarity, Adjusted Cosine Similarity, Pearson Correlation Coefficient (PCC), Constrained Pearson Correlation Coefficient, Sigmoid Function based PCC, Jaccard Coefficient, Mean Square Difference, PIP (Proximity-Impact-Popularity). (Matsunami, Ueda, & Nakajima, 2018) calculated similarity of users purchasing cosmetic products, based on online reviews provided by them. A new similarity measure was proposed to utilize the aspect level information in the user reviews. Sentiment score was calculated for each aspect resulting in large size of the rating matrix as compared to the traditional m (product) x n (users) size.

Table 2.2 lists the most common similarity measures in literature along with their advantage(s) and disadvantage(s):

Table 2.2: Advantages and Disadvantages of Similarity Measures

S.No.	Similarity measure	Advantage	Disadvantage
1	Cosine Similarity	1. Supports high dimensional data 2. Focuses on the vectors' directional similarity and not vectors' magnitude	1. Dependency on co-rated items-Not suitable in sparse rating environment 2. Some users give few ratings or do not rate at all 3. Considers the angle between the vectors and not the difference in magnitude or size of the vectors 4. Difference in rating scale is not taken into account i.e. two users with polar opposite ratings can be strongly correlated as per cosine similarity
2	Adjusted Cosine Similarity	Rresolves cosine similarity's drawback i.e. the average rating of user is subtracted to consider the difference in rating scale	This measure doesn't consider the preference of user ratings

Continued on next page

Table 2.2 – continued from previous page

S.No.	Similarity measure	Advantage	Disadvantage
3	Pearson Correlation Coefficient	This measure is useful in cases where user bias or different rating scales of users exist.	1. Dependency on co-rated items 2. Some users give few ratings or do not rate at all 3. Not suitable in sparse rating environment as depends on co rated items 4. Simple Pearson correlation doesn't take into account the number of common users 5. Cannot distinguish between dependent and independent variable i.e. cannot identify the cause and effect variables
4	Euclidean Similarity	This measure is used when the magnitude of the vector has its significance in the chosen application area	Euclidean distance is unsuitable for high dimensional data
5	Jaccard Coefficient	1. Suitable for binary data 2. Compares the set of patterns	1. Doesn't include the absolute rating value only shows the similarity of sample 2. The coefficient is affected by the total number of users or items"

Continued on next page

Table 2.2 – continued from previous page

S.No.	**Similarity measure**	**Advantage**	**Disadvantage**
6	Manhattan Similarity	Supports high dimensional data	1. Less intuitive than Euclidean distance 2. As the shortest path is not followed in Manhattan distance (vectors move in right angle), the resulting distance is greater than euclidean distance
7	Spearman Rank-Order Correlation Coefficient	Solves the sparsity and cold-start problem by assigning the missing values a basic rank value	1. Useful when the data doesn't follow the normal distribution 2. Less accurate as not all the data is utilised 3. Sensitive to data error and discrepancies
8	Kendall's Tau Correlation Coefficient	Kendall's Tau has smaller gross error sensitivity and asymptotic variance for normal data.	Difficult to compute
9	Chebyshev Similarity	Suits application that allow unrestricted 8-way movement of vectors	1. It can only be used for particular problems such as logistics. 2. It can't be applied for any general-purpose problem like Euclidean can.

The similarity measurements indicated above are based on the user's star ratings and do not take textual evaluations into consideration. Using methods from Natural Language Processing (NLP), such as topic modelling, text embeddings and sentiment analysis, it is feasible to include text based information into recommender system (Barrière & Kembellec, 2018). These strategies may be used to extract significant characteristics from textual reviews and include them into the recommendation process, either as additional inputs for the similarity measures or as independent models that can supplement CF or CBF.

There are several measurements of similarity based on side information such as textual reviews, such as (Chen et al., 2015):

1. **TF-IDF based Cosine similarity**:
   This metric encodes textual reviews as vectors of word frequency-inverse document frequency (TF-IDF) scores and calculates their cosine similarity (Xu, Dutta, & Ge, 2018). This method requires the pre-processing of text in order to extract features, which may be time-consuming, costly in terms of CPU resources, and resource-intensive in terms of memory; may be susceptible to noise and irrelevant characteristics.

2. **Word embeddings based similarity**:
   This metric captures textual reviews as dense, low-dimensional vectors that capture the semantic meaning of words and computes their similarity using cosine similarity (Elnagar, Al-Debsi, & Einea, 2020). It is necessary to have a substantial quantity of text data in order to train robust embeddings; the model may not function well on short texts or in situations in which there is a limited vocabulary.

3. **Sentiment based similarity**:
   This metric takes into account the sentiment polarity of reviews (e.g., positive, negative, or neutral) and computes the similarity between them based on the overlap of their sentiment distributions (Zhao, Lei, Qian, Mei, & Member, 2018). Such methods may oversimplify the more complicated and nuanced thoughts that are conveyed in reviews.

4. **Topic based similarity**:
   This metric portrays textual reviews as distributions across topics (themes or ideas) identified from the text using topic modelling methods and then computes the similarity between them based on the overlap of their topic distributions (Lin, Shen, Chang, & Chang, 2017). Pre-processing of text and topic modelling of that content are required, both of which may be computationally costly and may need some level of expertise.

As seen above, selection of a similarity measure in collaborative filtering is an important task (Fkih, 2021). The efficiency of a recommender system depends on the similarity measure chosen. Ratingbased recommender system considers similarity measures based on commonly rated items by its users. Also, reviews-based recommender systems have established their supremacy over rating-based recommender systems (Chen et al., 2015). In the studied literature, recommender system that employs usage of reviews for user similarity comparison is deficient. A gap is created due to usage of reviews as side information for building user/item profiles and user similarity computation using the provided ratings. Hence, there is a need to define similarity measures that rely on user reviews.

2.5 SPARSITY

Recommender systems employ user feedback such as star ratings on textual reviews to build on user and item profile and similarity measures to identify similar users and items. But, such systems grapple with sparse data because very few ratings are provided by users as compared to the quantity of items that exist on the platform that makes use of such recommender system. A sparse matrix representation is shown in Figure 2.6. In order to deal with this sparsity, user reviews are used as side information. This helps in augmenting the user/ item profile and improves the recommendation process.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4				5
User 2			3		
User 3	4				
User 4			3	3	
User 5		1			1

Sparsity = number of non-filled cells/ number of filled cells

Sparsity = 17/25= 0.68

Figure 2.6: An Example of a User-Item Sparse Rating Matrix

However, just like ratings, the reviews provided by users are also less in quantity as compared to the items present in the system. Further, the reviews are incomplete as not all aspects or features of a product are reviewed by the user. This leads to a novel sparsity problem due to subjectivity of reviews. This problem needs to be addressed in order to improve the performance of collaborative filtering algorithms thereby improving the overall recommender system (Idrissi & Zellou, 2020; Singh, 2020; Yang et al., 2020).

2.5.1 Aspect-based sentiment analysis

In order to understand the underlying intent of a user from the review, it becomes imperative to decipher the actual sentiment. (Kadhim, 2019; Hartmann, Huppertz, Schamp, & Heitmann, 2019) worked on text classification dealt with assigning of predefined categories to textual data which can be in the form of documents, paragraphs, sentences or phrases. Whenever textual data is categorized to access its sentiment, it is categorized into three classes namely, positive, negative or neutral. Aspect-based sentiment analysis assess the data based on the aspects mentioned in the data and classifies into the said categories based on the user opinion towards the aspects.

Previously, SemEval 2014 workshop a dataset for laptops and restaurants had been created, it contained aspect terms in the review sentences, their corresponding polarity, the category of the aspect and the corresponding polarity of each aspect category (Pontiki et al., 2015). SemEval 2015's workshop contributed dataset containing aspect category as a combination of entity and attribute type. SemEval 2016's workshop contributed multilingual datasets for restaurant (English, French, Spanish, Turkish, Russian and Dutch language), laptop (English), mobile phone (Chinese, Dutch), digital camera (Chinese), hotel (Arabic), museum and telecom (Turkish) (Hercigt, Brychcín, Svobodat, & Konkolt, 2016) domains. In the literature, there exists an Arabic book-review dataset with 14 aspect categories and 4 polarities (including polarity 'conflict') (Al-Smadi, Qawasmeh, Talafha, & Quwaider, 2015). IT item-review dataset also exists for the ABSA task (Tamchyna, Fiala, & Veselovská, 2015). The authors used machine learning approach to identify aspect terms and supplemented this process with hints from rule-based approach. (Sabeeh & Dewang, 2019) compared and classified Aspect-Based Sentiment Analysis (ABSA) techniques and pointed that an educated purchase decision does not require browsing of all the reviews. In order to identify aspects, earlier approaches have identified nouns as aspects. Also, probabilistic, association mining, clustering approaches have been employed for this identification.(Da'u, Salim, Rabiu, & Osman, 2020) used deep Convolutional Neural Network (CNN) to perform aspect-based opinion mining and tensor factorization to predict the overall rating. (Poria, Cambria, & Gelbukh, 2016) used seven layer CNN and word-embedding model for sentiment analysis for aspect extraction.(Noh, Park, & Park, 2019) also used CNN based model to develop aspect level classification model. There also exists contribution of non linear model, convolutional neural network (CNN) and statistical model Conditional Random Field (CRF) in the ABSA task (Da'u et al., 2020). However, ABSA is limited by limited covergae of lexicon based methods. ABSA requires large size of labeled training dataset for performing machine learning.

In the literature, ABSA has been performed on movie reviews, digital cameras, restaurants, telecom, consumer electronics, museum (Pontiki et al., 2015) on various languages such as Czech, Bangla (Rahman & Dey, 2018), French and Hindi (Akhtar et al., 2018). But, no annotated or labeled dataset, indicating the aspects and sentiments in a review, specifically for mobile phone domain in English language was encountered, the domain identified by the authors for their contribution. The growth of labeled datasets was observed to be less owning to the human involvement. A need for labeled dataset for performing aspect level sentiment analysis of mobile phone reviews given by customers in English language was identified. The contribution of such an annotated dataset would also help ML algorithms as part of supervised learning to predict the output, given the input text to be classified and the output label or aspect categories/sentiments.

A dataset of mobile phone (Apple-iPhone11) English reviews has been contributed, analysis of the statistical analysis on this dataset and result of state-of-the-art machine learning techniques on the collected dataset has been done. The goal was to determine the aspect category and aspect sentiment of the collected review texts through several machine learning methods. The different supervised machine learning techniques used are KNN, Logistic regression, Multi Layer Perceptron, Naïve Bayes, Random Forest, Support Vector Machine and a sequential model. In Sequential model, layers are stacked up on one another one at a time till the desired architecture is obtained. The first step is to provide the input features to the input layer and then decide upon the number of layers, neurons in those layers and the activation function. The next step is to compile the model for training the model. Training the model implies determining the best values of weight parameters to map input to the output over several iterations known as epochs. In this step the loss function for weight evaluation must be specified. Batch size can also be specified which corresponds to the size of the training samples within an epoch to be considered before the weight variables are to be updated.

2.5.2 Autoencoder

Although there are huge number of users and products available on e-commerce, very few users share their post experience through textual reviews leading again to sparsity problem in recommender systems (Idrissi & Zellou, 2020; Singh, 2020). This insufficient information affects inferring of user's preference which in turn hinders the performance of recommender system. Also, subjectivity of reviews leads to another type of sparsity problem in recommender system as all the aspects are not reviewed by the users (Yang et al., 2020). Techniques used to solve the sparsity issues in collaborative filtering systems are factorization machines, dimensionality reduction, clustering, co-clustering, Bayesian hierarchical modeling, transfer learning methods, ontology based methods, Bayesian personalized ranking, restricted Boltzmann machine and deep learning methods such as autoencoder collaborative filtering (Idrissi & Zellou, 2020). Due to the advancing capabilities of Artificial Neural Networks (ANNs) in providing increasing computational power, they have gained tremendous attention and have emerged as a trending filed in computer science (Batmaz et al., 2019). As deep learning models are highly effective in extracting hidden features and relationships and dimensionality reduction, they have been utilized in recommendation systems as well. Deep learning models used in recommendation tasks are restricted Boltzmann machine, deep belief network, autoencoder, recurrent neural network and convolutional neural network. Autoencoder (AE) models are commonly employed models of deep learning used in recommender system followed by CNN and RNN models (Da'u & Salim, 2020).

Further, missing value imputation techniques can also be utilized to solve the sparsity problem. KNN Imputation, Last Observation Carried Forward (LOCF), Hot Deck Imputation, Miss Forest, Random Forest, DataWig and Multivariate Imputation by Chained Equation (MICE) are some of the techniques that exist in literature to handle this issue (Yang et al., 2020; Emmanuel et al., 2021). These techniques work on ratings data as input and do not consider the aspect information contained in the reviews to remove sparsity.

(Li, Yuan, Qian, & Shao, 2018) studied the purchase-review behaviour of online customers and defined a new method introducing aspects to explore the customer's opinions. (Wen, Ding, Liu, & Wang, 2014) focused on the sparsity problem in collaborative filtering recommender system. They incorporated cosine similarity with Matrix factorization technique to overcome this sparsity issue (Ardimansyah, Huda, & Baizal, 2018; da Silva et al., 2018). (Yun, Hooshyar, Jo, & Lim, 2017) proposed a recommendation system based on subjectivity analysis on purchase reviews.

Autoencoder (AE) has emerged as one of the most effective methods for identifying the primary features of data. AE is a form of neural network that performs unsupervised learning tasks such as dimension reduction, efficient coding, and generative modelling (Khan, Niu, Sandiwarno, & Prince, 2021). Examples of these tasks include: Image recognition, computer vision, and speech recognition. These are only few of the application domains in which AE has demonstrated its superiority in the process of learning latent feature representation.

An autoencoder, as shown in Figure 2.7, is trained to copy its input to output. It is a feed-forward network, which has a propensity to learn a distributed data representation. The primary purpose of this kind of neural network is dimensionality reduction. Practically, there exists just one hidden layer in between its input and output layer, and the hidden layers' representation is much more condensed than that of input and output layers. The AE model requires that the input and output layers be configured in precisely the same way. AE in this manner is trained using the precise data that has been provided into the input layer of the model. Just like conventional neural network, AE training is performed through usage of backpropagation. The only thing that differentiates these is the error which is calculated by contrasting the output with the input data itself. Recently, several variants of AE have been developed, such as the sparsed-autoencoder, denoising-autoencoder, and stacked-autoencoder, in order to support robust feature representation for a variety of diverse applications. The AE algorithm is particularly effective for dealing with noisy data and learning the complex and hierarchical structure of the input data from the noisy data. In general, the AE model may be used for the creation of recommender system by either learning lower dimensional features at the outer layer or filling in the missing values of the user-item rating matrix in the construction layer. However, the most significant problem with deep AE models is that they are unable to search for the optimal solution. In addition to this, the training phase of the AE model requires a significant amount of time spent computing due to the extensive parameter tuning (Zhang, Liu, & Jin, 2020).

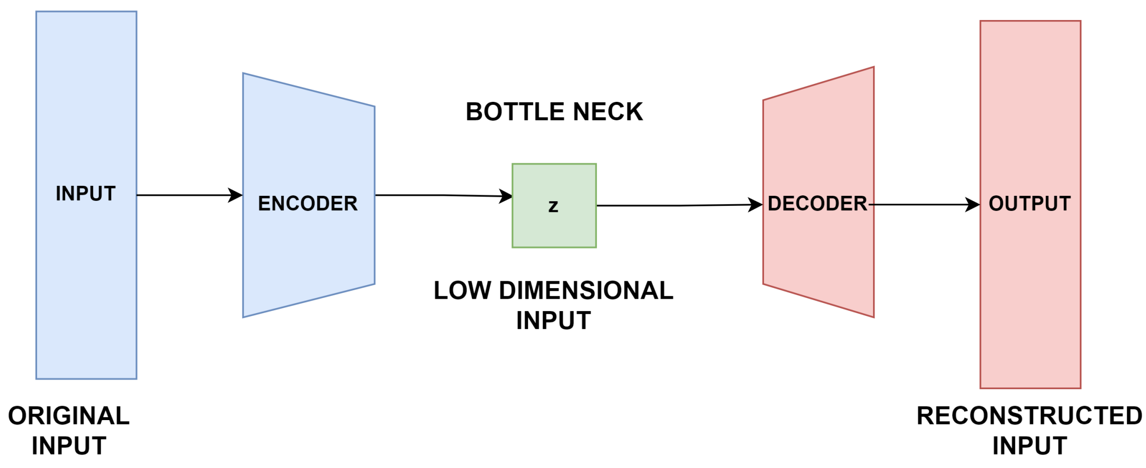

Figure 2.7: Illustration of Autoencoder Model

Recent changes made by AE to the recommendation architectures have helped re-imagine user experiences thereby enhancing customer satisfaction (?, ?). Recent studies have shown that combining other deep learning methods with AE-based recommendation systems has led to improved quality of recommendations. This is due to the fact that it is able to overcome the challenges that are presented by traditional recommendation systems while simultaneously producing high-quality recommendations. In recommender system that is based on AE, AE learns the non-linear user-item relationship and encodes these complex relationships into data representations, thereby better assisting the system in understanding. This allows the system a better understanding of both users and items. Additionally, AE is able to mitigate the effects of data sparsity by learning meaningful insights from various data sources like contextual, textual, and visual information.

Mostly, traditional recommender system focuses on a particular data source, such as ratings or textual information, as their primary concern. However, AE-based recommender system is capable of handling heterogeneous data sources, such as rating information, audio information, visual information, and video information. AE-based recommender system has a better understanding of the demands and features of items, in comparison to traditional recommender system, and AE-based recommender system achieves higher recommendation accuracy than traditional counterparts. AE-based recommender system also has a better understanding of the users' demands (Sedhain, Menony, Sannery, & Xie, 2015). When it comes to adaptability in multimedia contexts, AE-based recommender system outperforms the traditional ones. Further, the capability of AE-based recommender system to deal with noise is superior to that of traditional recommender system.

2.6 REVIEW ANALYSIS AND USEFULNESS

As a result of their ability to build trust in other potential customers within the online community, online user testimonials have attained a prominence that was very much required in the literature. Review usefulness is a measurement that indicates how beneficial a product review is to other consumers of the product. In other words, the helpfulness of a review is a metric that can be used to evaluate the quality of a review as well as its ability to influence the purchase choices of other users. When producing suggestions, many recommender systems take into consideration how useful a review was to the reader. This is because the helpfulness of a review may be a vital indicator of the relevancy and quality of a product. Analyzing the words of the review, looking at the reviewer's profile and previous actions, and compiling comments from other users are all standard methods for determining how beneficial a review is (Sun, Han, & Feng, 2019). The helpfulness or usefulness vote measure is often displayed next to the product review.

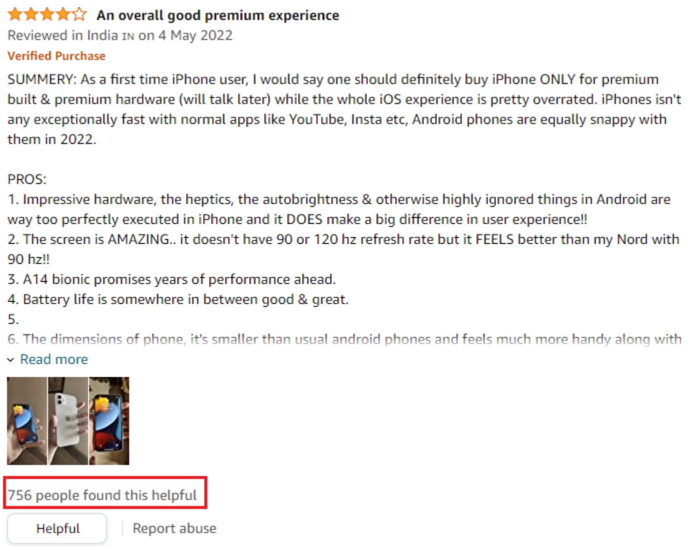

Figure 2.8: An example of a Mobile Phone Review with Helpful Votes

An example of an existing user review of Apple mobile phone that received 756 helpful votes from fellow users has been shown in Figure 2.8:

Using machine learning to make a prediction whether a review will be useful or not, there are a number of characteristics that are taken into consideration (Portugal et al., 2018). These parameters vary based on the application being used and the data that is readily accessible. The parameters can be majorly categorised into the following types:

1. **Review features**:
   Length of the review, sentiment of the review, presence of specific phrases, readability of the review and other linguistic factors fall under the category of review features. Also the review's metadata features such as date and time of review creation, and number of upvotes and downvotes are also categorised as review features.

2. **Reviewer profile features**:
   History of reviewer's helpfulness votes for a review, their indulgence or interaction level in the recommender system, their credibility and other user level features are categorized as reviewer profile features.

3. **Product features**:
   All the features related to a product such as its brand, price, validity, popularity, rating and domain etc. belong to product features category.

Machine learning algorithms are taught to predict the helpfulness of a review with a high degree of accuracy by taking into consideration the above relevant criteria. The prediction of whether or not a review will be useful can be viewed as a regression problem that assigns a score to each review. The scores can then be used to determine the review ranking or the recommendation(s). It is also possible to handle it as a classification problem in order to determine whether or not the review is of a high quality or useful.

(Enamul Haque, Tozal, & Islam, 2018) mentioned that product reviews can be viewed as a type of passive recommendation process or visibility of user sentiment for their past purchases.(Wu, 2017) in their work on review helpfulness found that the academic evidence on review usefulness is largely driven and aided by review hosting platforms, which offer users' opinions on reviews' helpfulness explicitly. (Malik & Hussain, 2018) reported that for instance on Amazon, customers do not only access the rating and text content of each user review, but they also view the number of votes the review obtains from the fellow users and the number of helpful votes. According to (Mitra & Jenamani, 2021), three qualitative perspectives namely, lexical, sequential and structural help to assess helpfulness of user reviews. They consider both semantic and syntactic features of review. (Hong, Xu, Wang, & Fan, 2017) stated that review's usefulness conveys the perceived importance of a review to the end-user. (Bilal et al., 2019) identified that this functionality, in particular, makes use of crowd-sourcing to assess the usefulness of reviews. Every review includes the question, "Was this review helpful to you?" Customers who read the reviews may upvote or downvote the review.

The research on review usefulness is roughly classified into two categories, prediction based techniques to ascertain the review's usefulness and understanding of review usefulness. (Dey & Kumar, 2019; Fan, Feng, Guo, Sun, & Li, 2019; Ge et al., 2019) used machine learning classifiers, regression and deep learning approaches to predict review helpfulness in the past. (Arif, Qamar, Khan, & Bashir, 2019) employed the usage of the review length, review timestamp, reviewer's expertise, and manner of writing reviews to predict helpful reviews.(Malik, 2020) identified review usefulness through earlier indicators namely, review length and review star rating. (Eslami, Ghasemaghaei, & Hassanein, 2018) found that the most useful reviews are said to be medium in length, have a lower score, and are negative or neutral in polarity.(Salehan & Kim, 2016) mentioned that besides the semantic aspect, neutral sentimental reviews are regarded to be also useful.(Krishnamoorthy, 2015) found the vital predictors of helpfulness as the inclusion of adjectives, status and action verbs, as well as grammatical structure, particularly when paired with readability and subjectivity, review age, and rating.Previous findings also indicated that the polarity of the review title, the sentiment and polarity of the product review, and the cosine similarity between the product review and the product title all contributed to the usefulness of user reviews. As per the literature, previous studies are deficient in terms of the combination of natural language processing tools and machine learning techniques for estimation of review usefulness. This study considering the above employs user voting as the target label to build the helpfulness or usefulness prediction system.

Prediction of review usefulness enables users to compose meaningful reviews that shall assist retailers in intelligent website management by guiding its users in purchase decisions (Malik, 2020). The incorporation of a usefulness estimation model can boost the effectiveness of a CF-based recommender system by optimizing the selection of appropriate data for estimation of user ratings. This is a great resource for identifying relevant user reviews for decision-making (Mauro, Ardissono, & Petrone, 2021). Table 2.3 lists the key takeaway points from the existing literature:

Table 2.3: Comparison of Existing Studies on Identification of Useful Reviews

SNo	Article	Model	Dataset	Input	Performance Metric	Key Points	C/R
1.	(Kong, Li, Ge, Ng, & Luo, 2022)	MLP, CNN with TransE	Amazon dataset:CDs, Electronics, Video Games, Books	Product, Review, Reviewer features	Accuracy, F1-score	Dependence solely on hand crafted features leads to poor accuracy. Along with CNN another technique is required for mapping between the reviewer, product and reviews	R

Continued on next page

Table 2.3 – continued from previous page

SNo	Article	Model	Dataset	Input	Performance Metric	Key Points	C/R
2.	(Bilal et al., 2021)	R:LNR, C:Log Reg, Both:DT, RF, GBT, NN	Yelp Shopping reviews	Product, Review, Reviewer features	RMSE, MSE, RAE, RSE, RRSE, MAE,R^2 and CC (R), Accuracy, AUC, Precision, Recall, and F1 score (C)	Authors examine the impact of friends on review usefulness by introducing social network features. For classification, reviews receiving more than 3 votes are marked as helpful, 0 votes as unhelpful and discarded otherwise	C, R
3.	(Du, Rong, Wang, & Zhang, 2021)	MLP, CNN	SiteJabber.com, ConsumerAffairs.com (Domains-Dating, Wedding Dresses, Marketplace, Car Insurance, Travel Agencies, Mortgages)	Review features	Accuracy	Adjacent or neighbour reviews impact a user's helpfulness perception of a review. For classification, reviews receiving more than 2 helpful votes labelled as helpful and unhelpful otherwise.	C

Continued on next page

Table 2.3 – continued from previous page

SNo	Article	Model	Dataset	Input	Performance Metric	Key Points	C/R
4.	(Mauro et al., 2021)	Linear Support Vector Regression, RF Regression	Yelp hotel stores reviews, Yelp food stores reviews	Review features	Pearson and Spearman correlation values	Deviations in star ratings, review's length and review's polarity w.r.t user and item impact usefulness. Authors do not consider reviewer features. Random Forest was a better helpfulness predictor. Integration of such an estimation model improves the CF system performance.	R
5.	(Malik, 2020)	Multivariate adaptive regression, 'C' and 'R' tree, RF, NN, deep NN	Amazon multi-domain sentiment analysis dataset	Review, Reviewer, Product features	MSE, RMSE,RRSE	Review type characteristics standout as effective characteristics to determine review's helpfulness as compared to reviewer and product characteristics. Combining all three characteristics yield best performance.	R

Continued on next page

Table 2.3 – continued from previous page

SNo	Article	Model	Dataset	Input	Performance Metric	Key Points	C/R
6.	(Akbarabadi & Hosseini, 2020)	DT, RF	Amazon Product dataset (Books, Office Products)	Review, Reviewer features	Accuracy, F-measure	Helpfulness threshold ratio set to value of 0.6. Features such as text, reviewer, readability perform better than summary features. RF performed better than decision trees	C
7.	(Malik & Hussain, 2020)	MLP, CART, Multivariate adaptive regression, Generalized Linear model, Ensemble model	Contributed dataset of 34 product categories from Amazon.com	Review, Reviewer, Product features	MSE, RAE, RMSE, RRSE, MAE	More the comments, polarity and sentiments in a review, more are the helpful votes. Reviews with atleast 10 votes are selected. Best results were obtained using hybrid features with ensemble model performing the best	R

Continued on next page

Table 2.3 – continued from previous page

SNo	Article	Model	Dataset	Input	Performance Metric	Key Points	C/R
8.	(Sun et al., 2019)	RF	Dataset from JD.com	Review features-informativeness, length	Accuracy, AUC	Classification threshold for search and experience products to be different. Threshold of 4 for search products such as electronics and 2 for experience products such as skin gave the best model performance	C

In Table 2.3, AUC stands for Area Under the Curve, 'C'-Classification, CNN-Convolutional Neural Network, CC-Correlation Coefficient, DT-Decision Tree, GBT-Gradient Boosted Tree, Log Reg-Logistic Regression R-Regression, RAE-Relative Absolute Error, RF-Random Forest, RMSE-Root Mean Square Error, R^2-R Squared, RSE-Relative Squared Error, RRSE-Root Relative Squared Error, MAE-Mean Absolute Erro, MLP-Multi Linear Perceptron, MSE-Mean Squared Error and NN-Neural Network.

2.7 REVIEW SUMMARY

A critical study of the above literature helps to understand the existing challenges faced by recommender system. In order to design a recommender system that incorporates product reviews instead of star ratings, deals with review-based similarity measures, handles associated sparsity problem and predicts useful reviews, this chapter surveys existing similarity measures, methods to remove sparsity, recommender system based on user reviews and prediction of useful reviews through product, review and reviewer features.

The study highlights that existing similarity measures are based on star ratings and have their own set of disadvantages for their usage in recommender system.

Similarity measures built on product reviews are found to be deficient in literature. Consequently, a novel method to identify similar users based on product reviews has been proposed. The study also reveals that sparsity is one of the major challenges of recommender system for which side information like user reviews are employed. However, user reviews lead to a new type of sparsity problem owing to their subjectivity. Sparse input data affects the performance of algorithms of recommender system. Aspect-based sentiment analysis of product reviews help to analyse the users' preference towards a products' features. Annotated dataset of mobile phone reviews for performing aspect level analysis is non existing in literature. This puts forth a requirement to build a labelled dataset containing aspect and its associated sentiment. Methods to mitigate the sparsity arising due to missing aspect preference information are discussed. Autoencoders are found to be the most widely used neural networks for alleviating the sparsity from the product-attribute perspective.

Lastly, the chapter emphasizes the importance of review usefulness in recommender system. A comparison of existing studies on identification of useful reviews has been presented. Prediction of useful reviews is considered as a classification or regression problem. The comparison concludes that dependence on only hand crafted features leads to poor accuracy. Review type features standout as effective characteristics. The threshold to label a as review useful or not useful varies in the literature and is decided through experimentation.

CHAPTER 3

RESEARCH OBJECTIVES

CHAPTER III

RESEARCH OBJECTIVES

This chapter briefs the research objectives identified after analysis of related research work and literature survey

---

3.1 GENERAL

As seen in the previous chapters, recommender system is an important means to filter information relevant to the users. Its existence is crucial to support decision systems. However, recommender system grapple with numerous challenges due to their high dependence on user and product profiles. Utilizing side information such as product reviews can help to deal with the existing flaws thereby bringing performance improvement to recommender system. The goal of this study is to handle the identified problems and provide a design of a recommender system using data analysis. The domain identified to proceed with the research work is e-commerce. Within e-commerce, mobile phone category has been identified as the target product of study. This selection is based on a user survey to determine the products that users are more likely to buy online through e-commerce platforms than offline. The problems can be majorly identified as review analysis incorporation, review-based similar user identification, data sparsity due to review-based subjectivity, absence of labelled dataset to perform aspect-based sentiment analysis, review usefulness and a study of impact of pandemic on e-commerce. Review analysis has been performed through usage of Latent Dirichlet Allocation (LDA) topic modeling technique and sentiment analysis. Data sparsity has been alleviated through aspect-based sentiment analysis, matrix factorization and autoencoder techniques. Machine learning algorithms have been used to predict the useful reviews. Emotional and sentiment analysis of users' social media interaction has been performed to analyse the effect of a pandemic on the e-commerce consumers.

The problems covered in this research are shown in Figure 3.1.

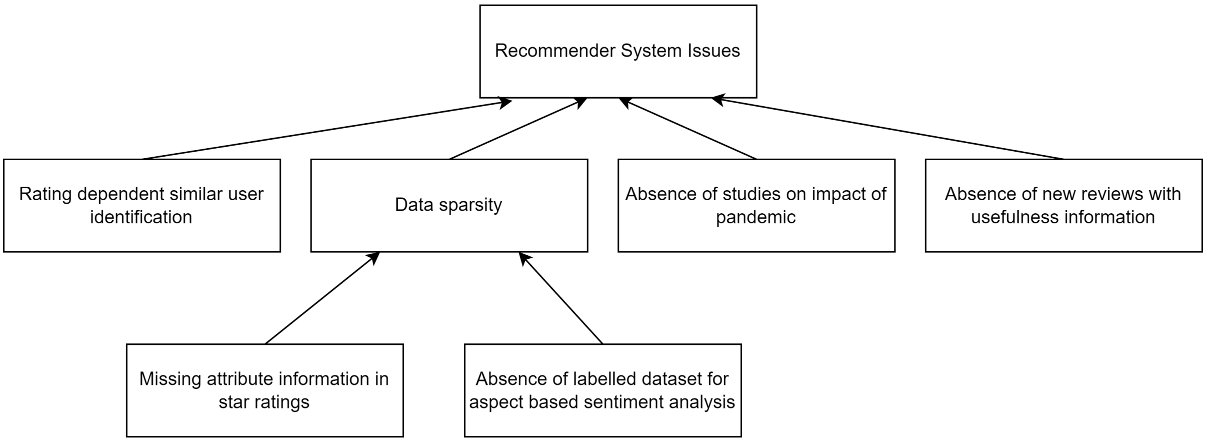

Figure 3.1: The Problems Covered in the Research Work

3.2 RESEARCH OBJECTIVES

The specific objectives of the research work are as follows:

1. ***To identify similar users based on customer reviews:*** *Existing research studies focus on identification of similar users through product ratings. Building customer and product profiles through customer reviews and identifying similar users through product ratings creates a gap that requires modification of existing process.*

   Solution: As part of this objective, the proposed method finds similarity between users based on the reviews provided by them for a product. The idea is to populate a list of similar users i.e. if two users provide similar reviews for similar products then these users are similar. Sentiment analysis of user reviews has been done to propose new measure to identify similar users.

2. ***To alleviate sparsity problem in recommender system:*** *Data sparsity is a well known problem faced by recommender system. This issue affects their performance. Side information such as customer reviews has been used in the past to tackle the sparsity problem. However, due to subjectivity of reviews, a new sparsity problem is generated. There is a need to alleviate this novel sparsity problem to improve the recommender system performance.*

   Solution: A new sparsity problem arising due to subjectivity of reviews has been identified and mitigated using techniques such as matrix factorization and autoencoder. User and item profiles have been built after taking into consideration aspects' importance. Also, a manually annotated dataset has been built to perform aspect-based sentiment analysis which forms the basis of quantification of aspect importance for a user or a product based on the provided reviews.

3. ***To determine the useful reviews for users:*** *In order to arrive at a purchase decision, users leverage the information contained in product reviews. However, browsing of large number of reviews is required to avail the benefit. Also, reviews are associated with users' vote to indicate their helpfulness. However, newly posted reviews do not receive these votes. Additionally not all reviews are tagged with such kind of votes by the users due to factors such as humongous volume of electronic word of mouth, voluntary helpfulness voting mechanism, level of visibility and their recentness. This necessitates the need to predict the review usefulness.*

   Solution: Feature engineering has been performed to derive review features. Useful or not useful review has been set as the target variable. The prediction problem has been treated as a classification problem. Several machine learning algorithms have been modeled, tested, tuned, compared and contrasted to predict the useful reviews. The threshold to label a review useful has been decided experimentally and set as ten in this study.

4. ***To incorporate reviews' analysis to generate recommendations for the user:*** *Attribute-level information is not available in star ratings. As a result, the user/product profile built through the help of user reviews can be used to augment the available product ratings. Also, interest of the user towards a product/service can be gauged in terms of the product's/service's features. This feature-based information is not available in recommender systems based only on product ratings. Hence, there is a need to build review-based recommender system to capture attribute level information.*

   Solution: Topic modeling technique helps to identify hidden topics of a document. Extraction of noun words in pre-processing helps to identify product features through topic modeling. Mapping of positive, negative and non reviewed features with overall features helps to identify uninteresting features. Exclusion of these features helps to downsize the recommendation list.

5. ***To analyse the degree of user inclination towards e-commerce:*** *During imposed lockdowns in the middle of pandemic, all domains functioning including e-commerce got stalled. As a result, customer interaction got minimised. As no customer feedback and existing studies on impact of pandemic were available, framing of policies and taking corrective measures for the benefit of all*

the stakeholders was delayed. This gave rise to study the impact of pandemic on e-commerce domain.

Solution: Tweeters resorted to usage of social media platform to express their mind on the imposed lockdowns by Indian government to confine the spread of the virus. An attempt has been made to understand the mind-set of Indian people using Python and R statistical software, during the pandemic. Also, opinion on e-commerce during this pandemic has been analyzed.

3.3 CHAPTER SUMMARY

This chapter briefs the objectives fulfilled as part of this study. The research objectives have been described along with their solutions. In the following chapters, the solutions to the identified objectives have been explained.

CHAPTER 4

A NEW APPROACH TO IDENTIFY SIMILAR USERS BASED ON CUSTOMER REVIEWS

CHAPTER IV

A NEW APPROACH TO IDENTIFY SIMILAR USERS BASED ON CUSTOMER REVIEWS

The chapter uses product reviews to identify similar users instead of conventional star ratings usage. A new approach has been proposed by computing the sentiment towards a product, list of related users and predicted sentiment.

4.1 INTRODUCTION

In order to provide a personalized experience to the user, collaborative filtering is the most used technique. The major step in this technique is to identify similar users or items with the help of user-product interaction matrix (Wang et al., 2017; Ayub, Ghazanfar, Mehmood, Alyoubi, & Alfakeeh, 2020). This is followed by generation of recommendations to the user(s). There are a number of pre-existing measures in literature to compute this kind of similarity, such as Cosine Similarity, Adjusted Cosine Similarity, Pearson Correlation Coefficient, Euclidean Similarity, Jaccard Coefficient, Manhattan Similarity, Spearman Rank-Order Correlation Coefficient, Kendall's Tau Correlation Coefficient, Chebyshev Similarity etc (Taghavi et al., 2018).

Cosine Similarity is found to be unsuitable in sparse rating environment. Also, while comparing, it does not consider the magnitude difference between two rating vectors. Strikingly, two users with polar opposite rating vectors are computed as strongly correlated as difference in rating scales is not considered in Cosine Similarity (Fkih, 2021). Pearson Correlation Coefficient does not consider number of common users in its computation and also can not identify the dependent and independent variables. Euclidean Similarity is unsuitable for high dimensional data. Jaccard coefficient computation is dependent upon the number of customers or items (Verma & Aggarwal, 2020). Spearman rank order correlation coefficient is useful for normally distributed data. Kendall's Tau is difficult to compute and Chebyshev similarity measure finds its application in logistic area. As these most commonly used measures suffer from several disadvantages and also do not make use of product reviews, a viable source of side-information, there arises a need for a measure based on product reviews (Saranya, Sudha Sadasivam, & Chandralekha, 2016).

Sentiment analysis of product reviews helps to determine users' attitude, emotions and sentiments towards a product. It is also known as opinion mining or opinion extraction. The goal is to identify the text polarity as positive, negative or neutral.

For instance,

1. **Positive polarity**: "I love this sweatshirt!"
2. **Negative polarity**: "The zipper broke on this piece the first time i wore it!"
3. **Neutral polarity**: "Just as pictured."

This kind of analysis can be performed at document, sentence or phrase level. These levels are briefed as follows:

1. **Document-level sentiment analysis**: The entire document is analysed for a single positive, negative or neutral polarity.
2. **Sentence-level sentiment analysis**: A document is made up of sentences. Each sentence of the document is labelled with positive, negative or neutral polarity.
3. **Phrase or Aspect-level sentiment analysis**: Each sentence is made up of phrases or aspects. Polarity corresponding to each phrase is identified and each phrase is categorized as positive, negative or neutral polarized (Musto, Gemmis, & Semeraro, 2017).

The extracted information can then be utilized to determine the sentiments towards unreviewed products (Kaur & Mangat, 2017). Sentiment analysis aids the decision making process as it helps the user to arrive at the answers of questions like 'which mobile to buy', 'which mobile has the best camera quality', etc. It also helps in predicting the future trends by keeping a track of users' demands. Thus, the insights offered by sentiment analysis can be leveraged by potential customers to buy a product as well as the vendors to upgrade their products (Aljuhani & Alghamdi, 2019).

4.2 DATASET DESCRIPTION

To implement the proposed method a subset of dataset of women's e-commerce clothing reviews has been taken from Kaggle (https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews).
The dataset has a total of 23486 customer reviews and 10 feature columns. The 10 feature columns are as follows:

1. **Clothing ID**: An integer value representing a unique clothing item
2. **Age**: Age of the item's reviewer
3. **Title**: Title of the item review
4. **Review Text**: Customer feedback of the item
5. **Rating**: Rating of the item ranging from 1(lowest) to 5(highest)
6. **Recommended IND**: This column has a value of 1 if the item has been recommended by a customer. It has a value of 0 if the item has not been recommended by a customer.
7. **Positive Feedback Count**: Number of reviewers who rated the review as positive
8. **Division Name**: Item's category
9. **Department Name**: Item's department such as tops, dresses, bottoms, etc.
10. **Class Name**: Item's class such as trousers, jeans etc.

4.3 METHODOLOGY

In this proposed method, sentence level sentiment analysis has been considered. As the user ID and item ID were not available as part of the dataset, the same have been assumed and a snapshot of the assumed dataset is shown in Table 4.1.

Table 4.1: Snapshot of Kaggle Dataset

User ID	Item ID	Review Text
U1	1	I purchased these in taupe, mint, and coral. They are extremely comfortable and soft. They can be rolled up to 3 different lengths. I stayed in my regular size and the fit is great. I can see why they sell out so quickly. A must for summer season.

Continued on next page

Table 4.1 – continued from previous page

User ID	Item ID	Review Text
U2	1	These are some of the softest most comfortable shorts i own and wish i had them in more colors. i like that i an adjust the length of the cuff since it's not "hemmed" to a certain length. i ordered the pink/salmon color and they go with so much!
U3	2	I saw this romper online and knew i needed it as i love flannels and i love rompers. it's super comfy. i bought with the intention of wearing it out, not just around the house. i think in the fall it will be cute with high socks and boots but as i just got it, i've been opting for tights. lots of compliments so far. i got a small because i didn't want it to be too short, it fits well.
U4	2	I wanted to love this romper, but it just wasted right for me. i am 5'5", 135lbs, 34c, curvy/muscular frame and ordered size small. i may have liked it better with more room in the medium, decided to return. i still recommend trying this product, but it wasn't for me and my hips!
U5	3	Love everything about it but had to get a size bigger to be long an off/ to short for a mom. if i was 20 would be fine.
U1	3	I went out on a limb ordering this romper. it's not really in my "wheelhouse" the whole romper thing. but it was a home run! i was afraid it might look too young and that i couldn't "pull it off," but the long sleeves and overall style was perfect. and my twenty something nieces were obsessed with it as well. as far as fit, i'm 5'7" 134lbs and ordered a size 4. i have a long torso and was concerned because some people said it ran short there, but i didn't have any trouble with that. i just took.
U2	4	So cute and so adorable but too short for my body in size small. i'm going to try a size medium. fabric is a nice weight cotton. lining is good, sleeves are a not too tight.
U3	4	This romper is cute, well-made and true to size, but i haven't figured it out how to put this on without having someone tie the back ties. which pretty much means that going to the bathroom is not an option while you have this romper on. not sure what the designers were thinking here. i'm returning this one.
U4	5	Mine came smelling like gasoline. not sure why, but i would have kept it otherwise. it's a smell that will be really hard to get out. looks like the picture.

Continued on next page

Table 4.1 – continued from previous page

User ID	Item ID	Review Text
U5	5	I love this sweatshirt! i truly did not pay much attention to it on line but while in my local store one was returned and it caught my eye immediately as the flowers are embroidered in a nice substantial rope type yarn to give it a more demential effect. the torn holes here and there give the appearance of being the most loved garment in your closet...and it has become mine along with the jacket with the same embroidery...funny how 2 pieces i did not give a second thought about have become my li
U1	6	I tried this on in the ivory color because it was on sale and i thought it ”might” be sort of cute. a comfy, flowy, warmer to cooler weather transition top. little did i know how much i would fall in love with it! i tried it on over like 3 things in the fitting room, including a black strapless maxi dress i wore into the store, and it still looked great! it is comfortable, loose, and goes with pretty much anything (i’ve tried and still haven’t found anything it looks bad over). it’s not super dr
U2	6	Ordered navy in a medium and it is wide. sometimes that’s a good thing, but not this time. am short waisted and hoped this would be more fitted at the waist on me but it isn’t. it just looked frumpy with lots of see through parts. obviously it needs a cami but wasn’t attractive on me at all. seemed well made and probably better on someone taller.

As part of the proposed methodology to identify similar users based on customer reviews, User ID, Item ID and Review Text are taken as input and are processed by three modules namely,

1. **Sentiment score calculator**: This module processes the input fields to generate the sentiment of a given review.

2. **User similarity finder**: This module takes the output of previous module as input and generates a list of similar users.

3. **Sentiment score predictor**: This module also takes the output of the first module as input along with the output of the second module to generate predicted sentiment scores of user reviews.

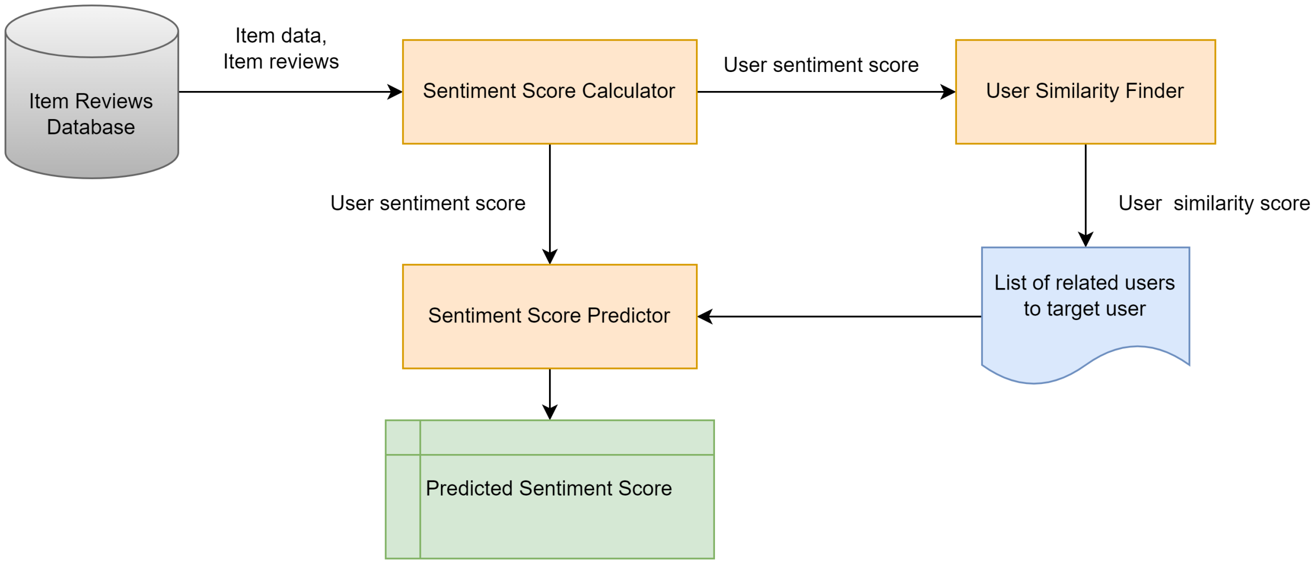

Figure 4.1: Proposed Method to Identify Similar Users through Product Reviews

For each item, the proposed method predicts a sentiment score for each user. The working of the proposed method has been represented through a diagram in Figure 4.1. The functionality of the proposed method is as follows:

4.3.1 Calculate sentiment score of a user for an item

The first step in the proposed method is to calculate the sentiment score for each review given by a user for an item. Sentiment score reveals the user's real experience with an item as it is based on the actual textual review provided by him. For this purpose, Python library *vaderSentiment* has been used to find the compound score for each feedback.
VADER stands for Valence Aware Dictionary and Sentiment Reasoner. It is a lexicon and rule-based open-sourced tool specifically designed for social media domain. It works with all other domains as well (Hutto & Gilbert, 2014). It is sensitive to polarity and intensity of the sentiments expressed in the reviews. The heuristics employed under it help to extend beyond the bag-of-words model. As it is a lexicon based approach, it consists of a list of words labelled as positive, negative or neutral based on their semantic orientation. Thus, it provides positive, negative and neutral composition percentage of each review along with a compound percentage. The compound score calculated by *vaderSentiment* on the range from -1 to +1 which has been mapped to a scale of 0-5. As the tool supports emoticons understanding and punctuation repetition, pre-processing of text was not required.

The calculated sentiment scores are shown in Figure 4.2.

```
I love this sweatshirt! i truly did not pay much attention to it on line but while in my local store one was returned and it ca
ught my eye immediately as the flowers are embroidered in a nice substantial rope type yarn to give it a more demential effect.
the torn holes here and there give the appearance of being the most loved garment in your closet...and it has become mine along
with the jacket with the same embroidery...funny how 2 pieces i did not give a second thought about have become my li 4.673

I tried this on in the ivory color because it was on sale and i thought it "might" be sort of cute. a comfy, flowy, warmer to c
ooler weather transition top. little did i know how much i would fall in love with it! i tried it on over like 3 things in the
fitting room, including a black strapless maxi dress i wore into the store, and it still looked great! it is comfortable, loos
e, and goes with pretty much anything (i've tried and still haven't found anything it looks bad over). it's not super dr 4.6985

Ordered navy in a medium and it is wide. sometimes that's a good thing, but not this time. am short waisted and hoped this woul
d be more fitted at the waist on me but it isn't. it just looked frumpy with lots of see through parts. obviously it needs a ca
mi but wasn't attractive on me at all. seemed well made and probably better on someone taller. 4.145

Sentiment Scores  [4.4025, 4.1, 4.631, 4.2764999999999995, 2.9295, 4.599, 4.494, 3.4815, 1.7845, 4.673, 4.6985, 4.145]
```

Figure 4.2: Sentiment Score of each Review

4.3.2 Finding user similarity

Based on the sentiment scores calculated above, user similarity is calculated. If the review provided by two users is similar i.e. if the sentiment scores calculated for two users are close to each other, then the two users are said to be similar, otherwise the two users are not similar. The calculation for user similarity using Python is shown in Figure 4.3.

```
  User ID  Item ID  Sentiment Score
0      U1        1           4.4025
1      U2        1           4.1000
For Item ID  1
Similarity between U1 and U2 is 4.6975
 -------------------
  User ID  Item ID  Sentiment Score
2      U3        2           4.6310
3      U4        2           4.2765
For Item ID  2
Similarity between U3 and U4 is 4.645499999999999
 -------------------
  User ID  Item ID  Sentiment Score
4      U5        3           2.9295
5      U1        3           4.5990
For Item ID  3
Similarity between U5 and U1 is 3.3305
 -------------------
  User ID  Item ID  Sentiment Score
6      U2        4           4.4940
```

Figure 4.3: User Similarity Score

The formula to find how much users are related to each other is given through Equation 4.1:

$$SimilarityScore(U_i, U_j) = \sum_{k=1}^{n} Senti_{max} - |Senti_{U_{ik}} - Senti_{U_{jk}}| \tag{4.1}$$

where, $Similarity(U_i, U_j)$ is the similarity value between two users U_i and U_j

. k is the number of items rated/reviewed by both U_i and U_j , $Sentimax$ refers to the maximum sentiment score for any review sentence i.e. 5 and Sentiment score for item k by user U_j.

For two users if there is no commonly reviewed item, then the user similarity score is zero for all such users. The user similarity score for the example dataset is shown in Table 4.2.

Table 4.2: User Similarity Scores for All Users

Similarity Score provided	U1	U2	U3	U4	U5
U1	1	9.144	0	0	3.305
U2	9.144	1	3.9875	0	0
U3	0	3.9875	1	4.645	0
U4	0	0	4.645	1	2.1115
U5	3.3305	0	0	2.1115	1

The similar users list is shown in Table 4.3 below.

Table 4.3: List of Similar Users

Users	Similar Users
U1	U2,U5
U2	U1,U3
U3	U2,U4
U4	U3,U5
U5	U1,U4

4.3.3 Predicting sentiment score of users for items

In this part, the sentiment score of each user for each item is predicted by using the sentiment score and similarity score found in above steps. The formula to predict user sentiment score is given in Equation 4.2 as follows:

$$E(y,k) = \sum_{U \in R} Senti_{Uk} + \frac{\sum_{U \in R} SimilarityScore(y,U)}{Senti_{max} * n} \tag{4.2}$$

where, $E(y,k)$ is the predicted sentiment score of user y for item k, R is the list of related users found in step 3.2, $Senti_{Uk}$ is the sentiment score of user U(in this case related user of user y), $SimilarityScore(y,U)$ is the user y's similarity score to user U and n is the total number of items.

Table 4.4 shows the predicted sentiment score using equation 4.2 for all the users.

Table 4.4: Predicted Sentiment Score for All Users

Users vs Items	I1	I2	I3	I4	I5	I6
U1	4.516	0.416	3.3455	4.91	5.01	4.561
U2	4.8405	5.06	5.03	3.9195	0.438	5.1
U3	4.718	4.8945	0.618	5.1	2.4025	4.763
U4	0.234	4.865	3.1635	3.7155	4.907	0.234
U5	4.5839	4.4579	4.7804	0.1814	1.9659	4.8799

4.4 RESULT & DISCUSSION

The user-item rating-based interaction matrix of the considered dataset is shown in Table 4.5. As seen from Table 4.4, the predicted sentiment scores are close to the provided star ratings.

Table 4.5: User-Item Rating Matrix Snapshot

Users vs Items	I1	I2	I3	I4	I5	I6
U1	5	-	5	-	-	5
U2	5	-	-	4	-	3
U3	-	5	-	2	-	-
U4	-	3	-	-	2	-
U5	-	-	4	-	5	-

For instance, user 1 had provided a star rating of 5 for item 1. The predicted sentiment score for item 1 by user 1 is 4.516. Similarly, user 2 provided star rating of 5 for item 2 and the predicted sentiment score for the same combination is 4.84. Likewise user 3 provided star rating of 5 for item 2 and obtained a sentiment score of 4.89. These approximations support the proposed similarity user calculation and predicted sentiments approach. Once the predictions are generated, the same can be utilised to suggest items of interest to the target users. That is, user 1 will be suggested items 4 and 5 as the sentiment score predicted is quite high. Likewise, user 3 will be recommended items 2 and 3, thereby achieving the objective of recommending items to users based on review-based similarity measure.

4.5 CHAPTER SUMMARY

In this chapter, a new approach to identify similar users by finding their sentiment for an item through textual reviews (English language) is proposed. Lexicon and rule based tool, *vaderSentiment* has been utilised to compute the user sentiment. The proposed system first calculates the user sentiment score for each item and then finds the user similarity with other users who have reviewed the same set of items. At the end, using both the above scores, the sentiment score for each item by each user is then predicted. This approach can be used to utilise the hidden sentiment stored in the form of text in user reviews as an input to collaborative filtering technique. However, as an improvement, this approach can be tested on complete datasets to verify method scalability.

CHAPTER 5

PROPOSED METHOD TO MITIGATE REVIEW BASED SPARSITY IN RECOMMENDER SYSTEMS

CHAPTER V

PROPOSED METHOD TO MITIGATE REVIEW-BASED SPARSITY IN RECOMMENDER SYSTEM

This chapter details the proposed solution to alleviate review-based sparsity problem in recommender system. Aspect-based sentiment analysis, user and product profiling, sparsity removal through matrix factorization and autoencoder has been performed. State-of-the-art recommendation system algoritms have been compared and contrasted for their performance with sparse and non-sparse input matrices.

5.1 INTRODUCTION

Since its beginning in the mid 1990s, recommender systems have been the subject of research in both the business world and academic institutions. It is a problem-rich study topic because of its practical applications including user participation, such as in the form of tailored product or service suggestions and the reduction in the number of available alternatives (Adomavicius & Tuzhilin, 2005). Recommender systems participate in many different platforms including Amazon, Netflix, eBay, Spotify, YouTube, and TripAdvisor They focus on both simple and complex things like books, movies, and computers (Jiang, Duan, Jain, Liu, & Liang, 2015; Bunnell et al., 2019). Sparsity is a concern that has consequences for both the system and the users. Although recommender system is put to commercial use, the system struggles with a lack of information due to the small number of evaluations from customers relative to the number of items and users. As a result, the recommender system's efficiency is diminished, as the similarity calculations between items or users are affected due to lack of information (Jiang et al., 2015; Taghavi et al., 2018).

The sparsity problem in recommender systems can be alleviated with the use of product reviews (Chen et al., 2015; Ahmadian, Afsharchi, & Meghdadi, 2019). It utilises auxiliary data by expressing the user's experience with the product's tangible and intangible attributes. Sentiment analysis is a key sub-field of Natural Language Processing (NLP) in which explicit or implicit opinion of users for a product are deciphered. The user's preferences are identified, which in turn motivates a sale. In contrast to a rating system, a review-based system can capture the nuanced perspective of its users. Aspect Based Sentiment Analysis (ABSA) collects the user opinion stated about a product's aspect at a fine level, adding value for the stakeholders that use the data (Tao & Fang, 2020). There has to be availability of labeled or annotated data for the models to be trained on in aspect-based sentiment analysis. ABSA techniques may be broken down primarily into lexicon and machine learning approaches, with the latter needing an annotated dataset for model training in supervised machine learning approaches, which is both time consuming and labour expensive. The sparsity problem first arose when consumers rated fewer goods and users than were really present in the system (Feng, Liang, Song, & Wang, 2020). Predicting the missing ratings helped recommender systems function better. The solution to this issue can be found in the user feedbacks used as supplementary data. Yet, using consumer feedback raises a new type of sparsity issue. The subjectivity of reviews, i.e. the limited product characteristics discussed in user evaluations, is the root cause of this new issue. The three main types of solutions for solving sparsity issues are matrix factorization, mathematical calculation, and computational intelligence. Nevertheless, these methods work best with a rating matrix between users and items, and not with aspect-based models (Yang et al., 2020). Because of this, Aspect-Based Collaborative Filtering (ABCF) has been introduced, a sparsity mitigation recommendation strategy that uses user feedback to boost suggestion quality.

5.1.1 Matrix factorization

Matrix factorization has been implemented and contrasted with autoencoder technique to mitigate the sparsity problem identified as part of this study. Let,
U - set of users,
I - set of items,
R - user-item rating matrix and
K – hidden features
Then, the task is to find $P(U * K)$ and $Q(I * K)$ matrix such that their product approximates R, i.e. R is approximately equal to $P * QT$. For obtaining P and Q, matrix can be initialized with some values and then the error value can be minimised

using gradient descent method iteratively.

$$(e_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} * q_{kj})^2 \tag{5.1}$$

Through Equation 5.1 missing values of R are predicted. Parameter alpha determines the rate of approaching the minimum. Regularization parameter beta is used to avoid overfitting. It controls the magnitude of the user-feature vector and item-feature vector. Including parameter alpha and beta to Equation 5.1 leads to Equation 5.2.

$$(e_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} (p_{ik} * q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^{K} (|P|)^2 + (|Q|)^2)) \tag{5.2}$$

In the proposed study, matrix factorization has been done for user-aspect importance matrix and product-aspect importance matrix to handle the missing values. User-aspect importance matrix is a preference matrix of users with product's aspects. Similarly, product-aspect importance matrix is a preference matrix of products with a product's aspects. The process has been repeated for 5000 steps with the value of alpha as 0.0002 and beta as 0.02. The updation of predicted values has been done according to the gradient descent formula. The Mean Squared Error (MSE) value has been set to 0.001 for termination of the factorization program. The non-sparse matrices obtained as a result of the factorization process have been then multiplied to obtain the user-product aspect weight matrix that will act as input to the baseline collaborative filtering recommendation algorithms.

5.1.2 Autoencoder

An autoencoder is a neural network which is feed forward type, meant to encode the input into a representation, followed by decoding of this representation so as to reconstruct the input. Generally, it is made up of three layers- input, output and hidden layer. While encoder part is represented by the input and hidden layer, decoder part is represented through output layer (Zhang et al., 2020). The neurons are numbered in a manner such that they are equal in the input and output layer. The input data $x = \{x_1, x_2, \ldots, x_n\}$which is dimensionally high is converted into a representation $h = \{h_1, h_2, \ldots, h_m\}$ which is dimensionally low and hidden. This is represented by a function f, shown in Equation 5.3:

$$h = f(x) = a_f(W * x + b) \tag{5.3}$$

where, a_f is an activation function. W refers $m * n$ a matrix for weights and

b refers bias $b \in R_m$. This hidden representation is mapped back to reconstructed $x' = \{x'_1, x'_2, ..., x'_n\}$ by decoder by a function g, shown in Equation 5.4:

$$x' = g(h) = a_g(W'h + b') \tag{5.4}$$

where, a_g is decoder activation function, W' is $n * m$ weight matrix and $b' \in R_n$ is the bias vector. The activation functions are usually non-linear for example hyperbolic tangent function and the sigmoid function. AE is known to minimize the reconstruction error. The reconstruction error is formulated using square-error or the cross-entropy error whose formulae are as provided in Equation 5.5 and Equation 5.6 respectively:
Square error:

$$E_{AE}(x, x') = (|x – x'|)^2 \tag{5.5}$$

Cross-entropy error:

$$E_{AE}(x, x') = -\sum_{i=1}^{n} x_i \log x'_i + (1 - x_i) \log(1 - x'_i)) \tag{5.6}$$

Loss function of AE can be constructed by adding regularized term to reconstruction error and is defined as follows through Equation 5.7:

$$L_{AE}(x, x') = \sum_{x \in R^n} E_{AE}(x, x')) + \lambda * Regularization \tag{5.7}$$

The loss function is optimized by Stochastic Gradient Descent (SGD) or the Alternative Least Squares (ALS). Typically, non-linearity in the user-item ratings is learned and reconstructed so as to generate the missing ratings. Training phase in autoencoders is faster due to usage of gradient-based backpropagation and it generates more accurate recommendations as it minimizes Root Mean Square Error, a commonly used evaluation parameter for recommender systems.

5.2 DATASET DESCRIPTION

The data used as part of this study refers to Kaggle's mobile phone reviews dataset "https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones". The dataset originally contains six columns namely, product title, brand name, price, rating, reviews and review votes. It contains 162,492 unique reviews made for 4410 mobile phone devices. The rating column is numeric, varying on a scale of 1 to 5. In this study, each product name has been assigned unique product ID, since the goal of recommender engine is to generate product recommendations.

Table 5.1: Description of Rating Column

Descriptive Statistics Parameters	Parameter value
count	413840
mean	3.82
std	1.55
min	1
25%	3
50%	5
75%	5
max	5

Table 5.1 consists of a description of the rating column in the given dataset. Table 5.2 displays the total number of reviews corresponding to each rating scale value. 2,23,605 reviews received the highest voting, 5, in the given dataset. Very few reviews, 24,728 received rating value 2 from the users.

Table 5.2: Dataset Description-Number of Reviews for each Rating Level

Rating	Review count
5	223605
1	72350
4	61392
3	31765
2	24728

Samsung, BLU, Apple, LG, Blackberry, Nokia, Motorola, HTC, CNGPD and OtterBox are ten prominent brands in the dataset according to the rating count received by them from the users. Out of these top ten brands, the maximum number of ratings is received by brand Samsung and the minimum number of ratings is received by Otterbox brand.

5.3 METHODOLOGY

The goal is to inculcate aspect level information from user reviews by extracting aspect terms, followed by their mapping to pre-defined aspect categories and perform aspect level sentiment analysis. The purpose of Aspect Based Sentiment Analysis (ABSA) is to determine the polarity of a review's aspect. This sort of sentiment analysis is more granular than earlier types, such as document level and phrase level sentiment analysis. (Noh et al., 2019). The following steps constitute the ABSA procedure (Kiritchenko, Zhu, Cherry, & Mohammad, 2015):

1. Identifying aspect terms in sentences

2. Identifying the polarity of aspect terms

3. Identifying of aspect classes

4. Identifying the polarity of aspect classes

5.3.1 Manual ABSA

ABSA consists essentially of two tasks: the first is to identify the elements of the assessed product, and the second is to discover the user's sentiments towards these aspects. The entities for which ABSA has been performed are movie reviews, digital cameras, restaurants, telecommunications, consumer electronics, and museums in a variety of languages such as Czech, Bangla, French and Hindi (Pontiki et al., 2015; Rahman & Dey, 2018; Akhtar et al., 2018) . However, no labeled dataset exists in the literature for the mobile phone domain in the English language. As the growth of labeled datasets is dependent on human intervention, it becomes critical to contribute such annotated datasets (Kou et al., 2020). Tagging aspect categories entails identifying an aspect category and assigning it to a review sentence. For example, "The iPhone design is good and the camera quality is awesome" has two aspect categories: Mobile Design and Camera Quality. The polarity label for both of these categories will be positive in this example. An annotated dataset has been created for performing aspect-based sentiment analysis of mobile phone reviews provided by Amazon's customers in the English language. The phone under consideration is Apple's iPhone11.

Labeled dataset is required for execution of supervised machine learning algorithms. The contribution of the paper is a dataset of mobile phone reviews from Amazon India which has been manually tagged for aspect categories and aspect sentiments by a team of 6 people including the authors. The prepared dataset is novel as no other dataset for performing supervised machine learning for aspect based sentiment analysis of mobile phone reviews in English language is available. This dataset can be used in the area of recommender systems to understand the mindset of customers towards the aspects of mobile phone. 960 user reviews of the black-colored 64GB variant have been downloaded through Python. These user comments have been collected in an excel file using Python's BeautifulSoup package that traverses the HTML parse tree to access web page elements. Once collected, the reviews have been divided into sentences, resulting in a total of 2109 review sentences. The resulting dataset's accuracy has been validated using state-of-the-art machine learning techniques. When a machine is trained to predict the output given the input text to be classified and the output label or aspect categories/sentiments, this is referred to as supervised learning (Al-Smadi, Qawasmeh, Al-Ayyoub, Jararweh, & Gupta, 2018; Portugal et al., 2018; Kadhim, 2019; Shaheen, 2019; Dragoni, Federici, & Rexha, 2019). The goal was to use deep learning and machine learning methods to determine the aspect category and sentiment of the collected review texts. The following are the various supervised machine learning techniques used:

1. **K nearest neighbor (KNN)**: KNN is an abbreviation for k-nearest neighbor, a statistical classification method. It is a non-parametric classifier from the family of proximity-based algorithms (Kou et al., 2020; Hartmann et al., 2019). In this method, the nearest neighbors of the labeled examples from the training review are ranked for each test review, and then a class assignment is derived using the categories of the highest-ranked neighbors. This model does not learn; instead, it memorizes and represents the entire dataset (Raza, Hussain, Hussain, Zhao, & ur Rehman, 2019). For high dimensional and sparse data, distance computation for the similarity between test and training reviews is computationally expensive. The most commonly used distance measures in this method are:

   (a) **Euclidean distance**: Euclidean distance between two points, as shown in Equation 5.8, X and Y is determined as square root of sum of sum of their squared differences across all input attributes *i* (Raza et al., 2019):

   $$Euclidean\, distance(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{5.8}$$

   (b) **Manhattan distance**: Sum of absolute difference of two points gives the Manhattan distance between two points, shown in Equation 5.9.

   $$Manhattan\, distance(X,Y) = \sum_{i=1}^{n}|x_i - y_i| \tag{5.9}$$

   (c) **Minkowski distance**: Generalization of above two distances gives the Minkowski distance, shown in Equation 5.10.

   $$Minkowski\, distance(X,Y) = (\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}} \tag{5.10}$$

where, value of p is either 1 or 2. Manhattan distance is obtained when i is equal to 1 and Euclidean distance is obtained when p is equal to 2. As a non-parametric method, this method suffers from the curse of dimensionality, requiring a large number of training examples to generalise satisfactorily for more features. Overfitting is more likely when large amounts of training data are provided in this manner. As a result, it is preferred for short texts rather than long texts (Vandic, Frasincar, & Kaymak, 2018).

2. **Logistic Regression (LR)**: In this method, given the input vector, the output class is assigned a probability (Raza et al., 2019; Rodrigues, Chiplunkar, & Fernandes, 2020). The Logistic Regression model is based on the logistic function or sigmoid function. An S-shaped curve maps real values to values between 0 and 1. The standard notation for the sigmoid function shown in Equation 5.11 is:

$$\frac{1}{1+e^{-z}} \tag{5.11}$$

   where, z is any real number to be transformed between 0 and 1. Logistic regression is a multi-class classification problem that began as a binary classification problem. For an input sample z, the probability of being in the first class is given through Equation 5.12 as:

$$p(x) = \frac{e^{\beta^T.z}}{1+e^{\beta^T.z}} \tag{5.12}$$

   where, β is the parameter vector and z is the training sample. During training, algorithms such as maximum-likelihood estimation are used to minimize errors in the predicted probability.

3. **Naïve Bayes (NB)**: Naïve Bayes is a generative probabilistic classifier that makes use of the properties of the Bayes theorem to hypothesize the relationship between independent variables. The training documents estimate the conditional probability P(d—c) of a document belonging to a class, and the test documents estimate P(c—d) using the Bayes theorem, as shown in Equation 5.13:

$$P(\frac{c}{d}) = \frac{P(c).P(\frac{d}{c})}{P(d)} \tag{5.13}$$

   It works well for independent features, which is also the method's underlying naive assumption. Due to the inherent regularisation, NB is less likely to overfit than discriminative classifiers and performs well for smaller samples. This method is incapable of modeling feature interaction. NB classifiers have three types: Bernoulli, Gaussian, and Multinomial. Gaussian is used for continuous datasets, Bernoulli is used for binary datasets, and Multinomial Naïve Bayes is used for count datasets (Hartmann et al., 2019; Aljuhani & Alghamdi, 2019). This classifier is used when memory and processing are important factors (Shaheen, 2019).

4. **Random Forest (RF)**: It is a discriminative classifier (Hartmann et al., 2019) based on multiple decision trees. A decision tree is made up of nodes and edges, where nodes represent the value of an attribute and edges represent the result of a test (Imtiaz & Islam, 2020). The best feature is chosen for splitting the node in the forest. For classification, the test is started at the root node and the edges are followed based on the results; the process is repeated until the leaf node is reached, and finally, the outcome corresponding to the leaf is predicted (Mohammed & Kora, 2023).

5. **Support Vector Machine (SVM)**:

   This is a discriminative classifier that attempts to identify a decision boundary by transforming non-linearly separable data to a higher dimension space with a separating hyperplane. The hyperplane can be represented through Equation 5.14 as:

   $$w.x + b = 0 \tag{5.14}$$

   where, w is the weight and b is the bias or the intercept. Each input point representing the sample lies on either side of the hyperplane. Initially designed for solving two-class problems, the decision surface separates the data points in the best manner with a maximum possible margin between the two classes. The data points which contribute to defining the margin are called support vectors. The magnitude of margin is the perpendicular distance from the hyperplane to the data points. The goal of training in SVM is to find the coefficients that separate the classes optimally, refer Equation 5.15.

   $$y_i = (w.x_i + b) - 1 \geq 0 \, \forall i \tag{5.15}$$

6. **Deep Learning (DL)**: Deep learning is a branch of machine learning that is inspired by neural networks. Deep learning models, as opposed to machine learning, learn the problem's features on their own without requiring it to go through a feature extraction process. Layers are stacked on top of one another in the Sequential model one at a time until desired architecture for Multilayer Perceptron is achieved(Chollet, 2020). The first step is to provide the input features to the input layer, after which the number of layers, the number of neurons in each layer, and the activation function are determined. The fol-

lowing step is to compile the model for training. Training the model entails determining the best weight parameter values to map input to the output over several iterations known as epochs. The loss function for weight evaluation must be specified in this step. The batch size corresponds to the number of training samples to be considered within an epoch before the weight variables are updated, can also be specified.

7. **Multi Layer Perceptron (MLP)**: The Perceptron was first introduced as a model of the biological neuron for binary classification (Raza et al., 2019). It has been generalized to deal with multi-class problems. In Perceptron, the input is mathematically transformed by multiplying the input by the weight parameter, summing the weighted inputs, adding the bias variable, and passing it to an activation function, which produces the final output (Varghese, Agyeman-Badu, & Cawley, 2020). The binary classification activation function is defined through Equation 5.16 as follows:

$$\phi(z) = 1\, if\; z \geq \theta, -1\, otherwise \tag{5.16}$$

where, z is the net input defined through Equation 5.17 as:

$$z = w_1 x_1 + w_2 x_2 + ..w_n x_n \tag{5.17}$$

where, x is a sample from the training set, w is the corresponding weight vector and θ is threshold. The algorithm begins by initializing the weight vectors with zero. The corresponding predicted class for each sample is computed and compared to the actual class value. If the predicted and actual class values differ, the weights are updated. The updated weight vector is defined through Equation 5.18 as follows:

$$(\tilde{w_j}) = w_j + \eta(y^i - \tilde{y^i})x_j^i \tag{5.18}$$

where, η is the learning rate, y^i represents the actual class and $(\tilde{y^i})$ represents the predicted class for sample $x^(i)$. Several layers guide classification in a Multi-layer Perceptron network (Imtiaz & Islam, 2020). It is termed as a logistic regression classifier variant. It is a subtype of the feed-forward artificial neural network where the features of the input data are transformed into a predefined number of linearly separable spaces, with each layer fully connected.

5.3.2 Manual ABSA- dataset description

The dataset constructed for ABSA consists of product reviews of the Apple-iPhone11 mobile written by Amazon customers in the English language. A survey of 21 questions and 202 respondents conducted by the authors was used to scrap the dataset consisting of 960 product reviews. The online survey was conducted to understand the users' preference for usage of e-commerce platforms for purchasing products online, consideration of product reviews in making their decision to purchase/reject a product, preference towards the brand and aspects/features of a mobile phone and their inclination for the operating system of a mobile phone. The questionnaire of the survey is added in the Appendix B for reference.

Apart from questions about the survey's objective, the survey includes questions about user demographic information such as age, gender, profession, and education level. The questionnaire responses assist in justifying the dataset preparation. Figure 5.1 displays responses to age group.

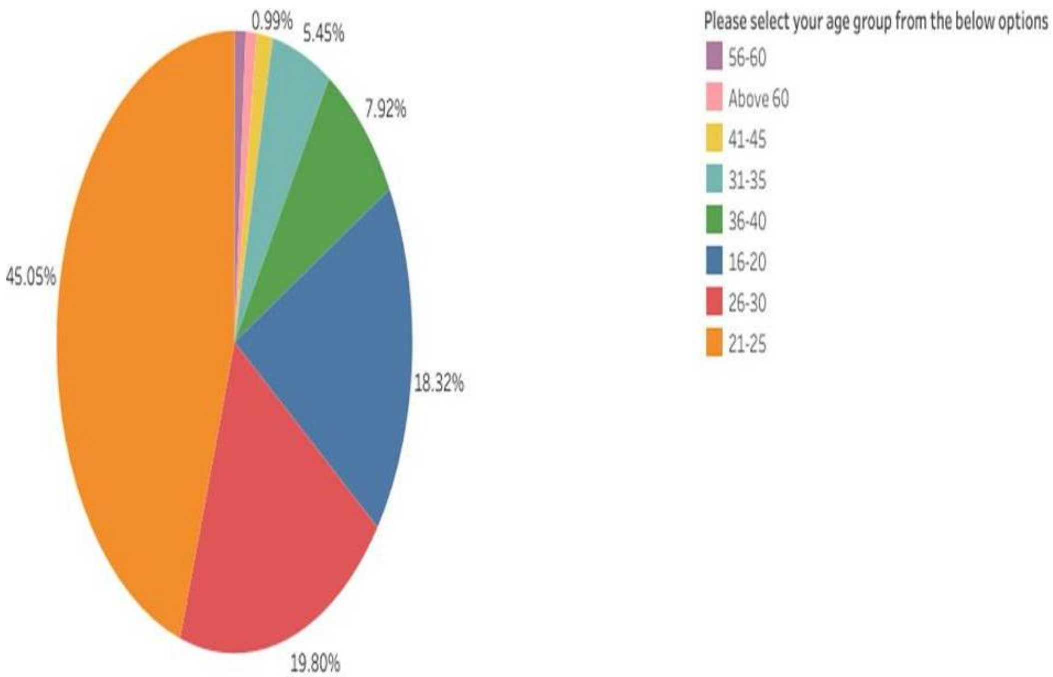

Figure 5.1: Survey Responses to Age Group

18.3% of respondents are between the ages of 16 and 20, 45% of respondents are between the ages of 21 and 25, 19.8% of respondents are between the ages of 26 and 30, 5.4% of respondents are between the ages of 31 and 35, and 7.9% of respondents are between the ages of 36 and 40.

From the total responses received, 61.9% are female respondents and 35.6% are male respondents as shown in Figure 5.2.

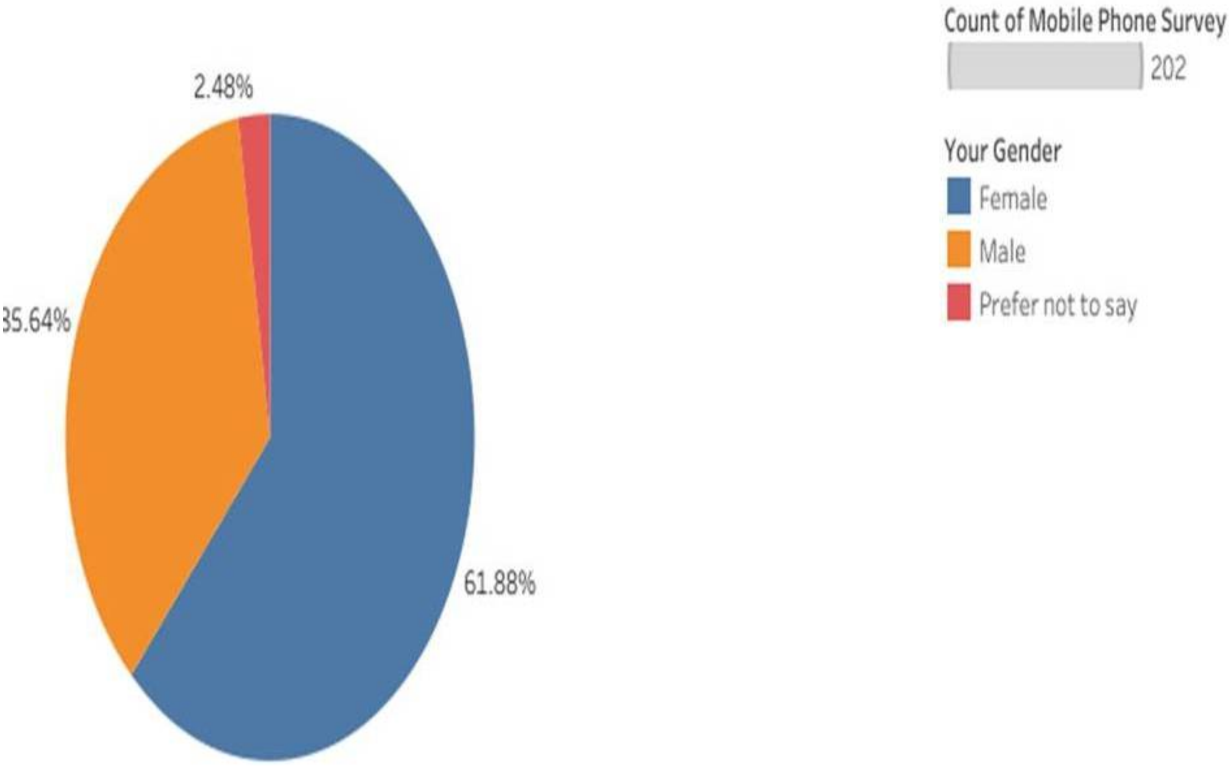

Figure 5.2: Survey Responses to Gender

It should be noted that there is no relationship between the questionnaire respondents and the product reviewers under consideration. The responses are collected using Google forms, and the reviews are obtained from the Amazon India website.

The majority of respondents (67.3%) are students, with only 16.3% working and 2% running their businesses. The majority of respondents (31.2%) own a Redmi phone, followed by Samsung (15.3%), OnePlus (11.9%), and Apple (11.4%). Approximately 65.9% of respondents have been using their current mobile phones for the past two years, while only 6.4% have been using it for more than four years. The majority of respondents (53%) are satisfied with their current mobile phones, and 81.2% think Android is a better operating system in a mobile phone.

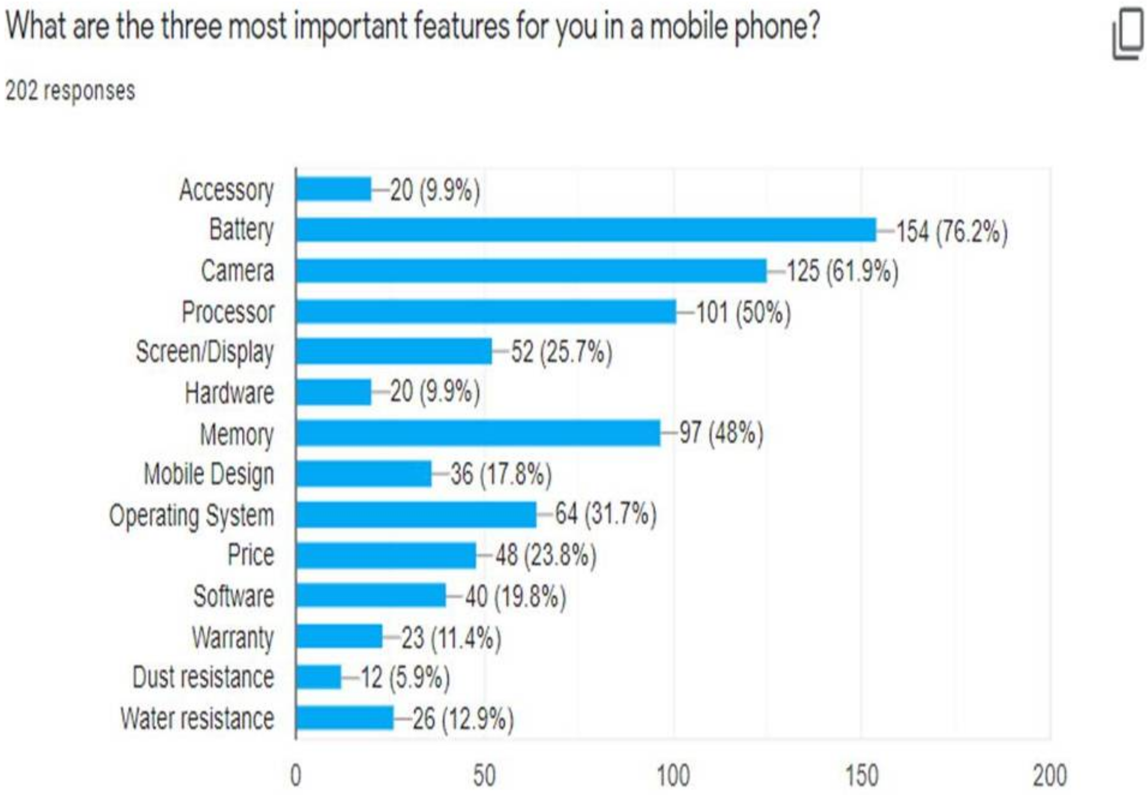

Figure 5.3: Survey Responses to Important Aspects of Mobile Phone fac

Figure 5.3 displays important aspects of mobile phone to users. 29.2% of all respondents have voted for Amazon's e-commerce platform as a better place to buy a mobile phone. Another 29.2% of respondents think that offline stores are a better option. Furthermore, Amazon and Flipkart have received 25.7% of the responses.

However, as shown in Figure 5.4, Flipkart has received only 11.4% of the responses.

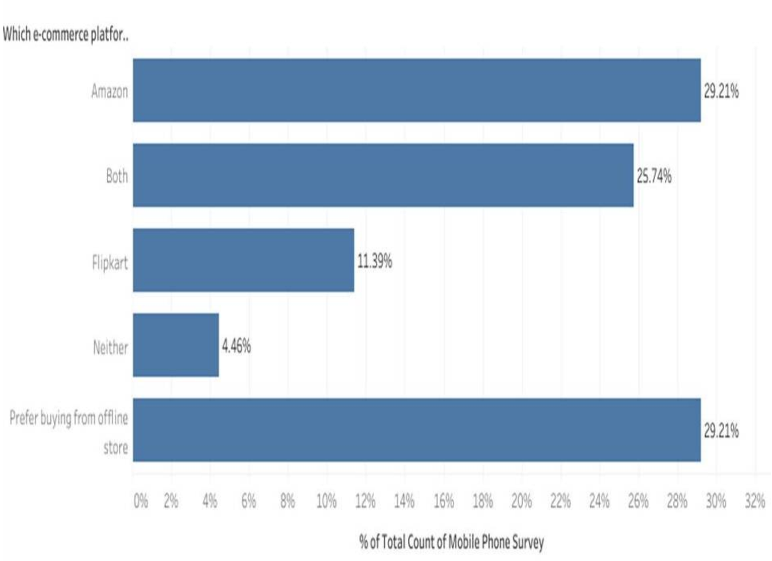

Figure 5.4: Survey Responses to Preference for E-Commerce Platform

70.3% of respondents consider a product's review before purchasing a product from e-commerce platforms, but only 23.5% provide the review after purchasing the product indicating the sparsity of data on user reviews.

The same has been illustrated in Figure 5.5.

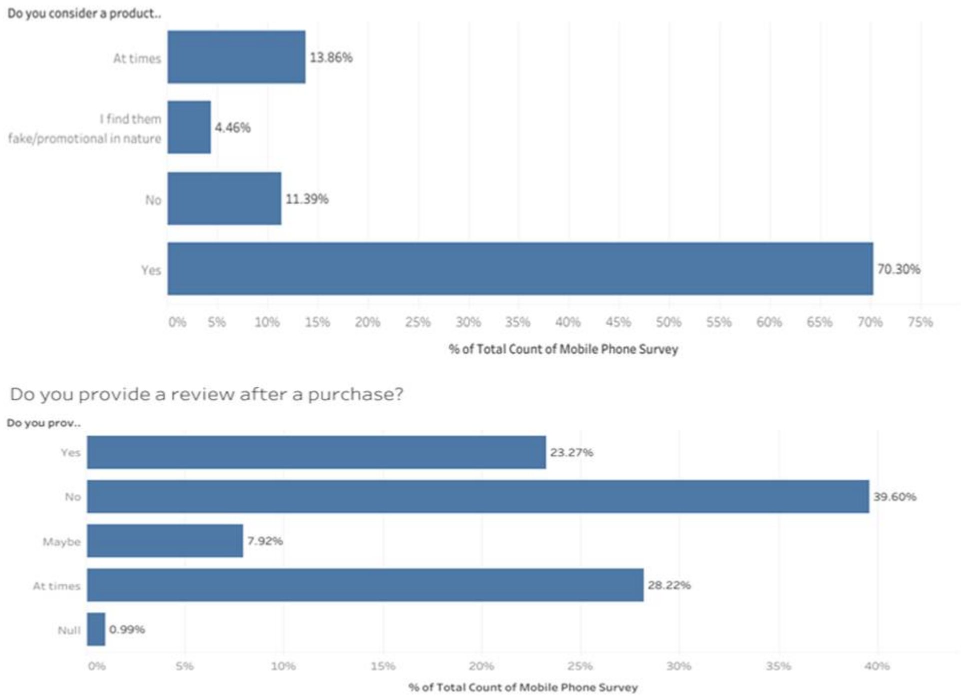

Figure 5.5: Survey Responses to Consideration of Product Reviews Before and After Purchase

As shown in Figure 5.6, when comparing user review and rating as to which one is more dependable, user reviews have received 18.8% of the responses alone and user ratings have received only 5.9% of the responses. Together, reviews and ratings received 62.4% of the responses.

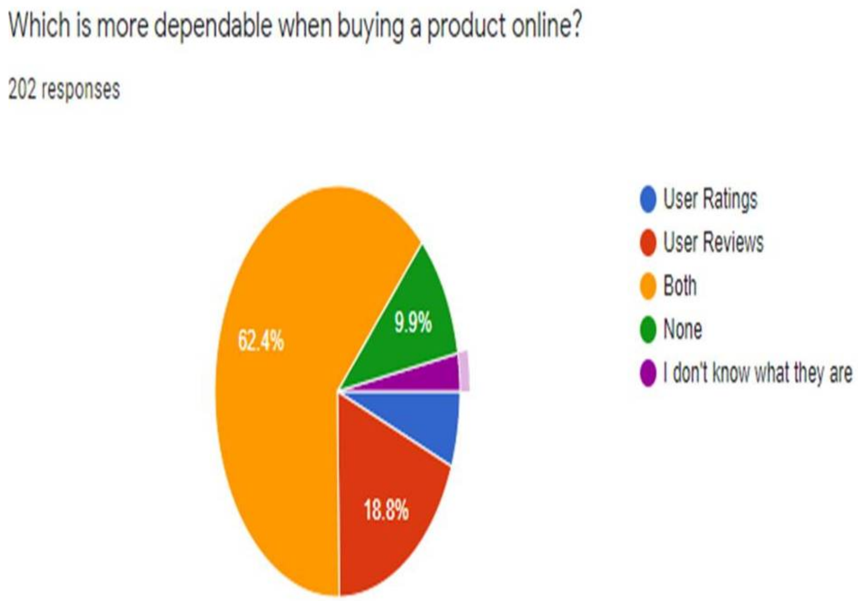

Figure 5.6: Survey Responses to Dependability on User Reviews vs User Ratings

Lastly, 32.7% of respondents think Apple has the best phone on the market right now, but only 11.4% of the total own it,, as shown in Figure 5.7

Figure 5.7: Survey Responses to Best Mobile Phone Brand

Apple-iPhone11 reviews have been selected to prepare the dataset as the majority of the users think Apple is the best phone manufacturer. The dataset for ABSA of mobile phone reviews in the English language is created programmatically by scraping publicly available reviews from Amazon's Indian e-commerce site of Apple-iPhone11. Following data collection, the dataset is annotated with a predefined set of aspect categories.

5.3.3 Manual ABSA-data collection

The scraped dataset includes 960 mobile phone reviews (2542 sentences). For training and testing the ABSA system, the dataset is cleaned manually to remove review sentences with no semantics. As a result, a total of 2109 sentences have been manually annotated with relevant aspects/features and sentiment categories by annotators. The curated dataset is available for reference in Appendix C. Figure 5.8 depicts the ten most frequently occurring words in the scrapped dataset.

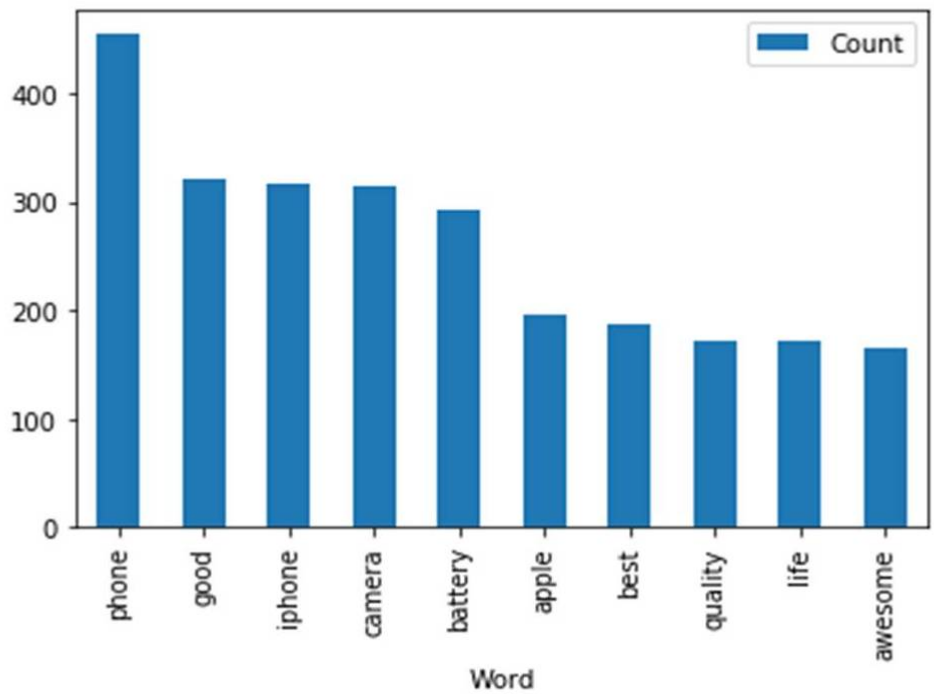

Figure 5.8: Top Ten Commonly Occurring Words in Reviews

Each review has been broken down into sentences using a full stop as the sentence terminator programmatically. A common review ID has been assigned to all the sentences of a review. Sentences with fewer than three alphabets are eliminated. Because review sentences can have multiple aspects, such sentences have been repeated and different aspect categories have been tagged for each of them, as shown in Table 5.3:

Table 5.3: Sample Annotated Dataset

S.No	Review Text	Aspect Sentiment	Aspect Category
1	28/09/19, but the thing I got started heating up every now and then	Negative	Performance
1	As it continued, tried to return the product by speaking to Amazon customer support but in vain	Negative	Amazon
1	Contacted Applecare, just to be consoled that it's quite normal	Neutral	Brand
1	I was much elated to receive the iPhone 11 so fast, next day of dispatch i.e.	Positive	Delivery
1	It was handed over to the Apple ASP as the return window closed on 10/10/19 (what use it was for??) and diagnosed as having issues and has further been sent to Apple repair facility at Bengaluru	Negative	Brand
1	So I'm here w/out my first iPhone after using it(suffering for??) just a little over 2 weeks and the CREDIT GOES TO AMAZON !! Bravo, keep it up Amazon	Negative	Amazon
1	Some body called me back to convey that only Apple will decide which one to take back	Negative	Amazon
1	Why is then Amazon took up the sacred duty of selling such an item which they can't exchange/ have no control ? The product developed new issues like proximity sensor malfunction and last but most importantly loosing mobile network every other minute(even had two software updates)	Negative	Amazon
1	Why is then Amazon took up the sacred duty of selling such an item which they can't exchange/ have no control ? The product developed new issues like proximity sensor malfunction and last but most importantly loosing mobile network every other minute(even had two software updates)	Negative	Hardware

Continued on next page

Table 5.3 – continued from previous page

S.No	Review Text	Aspect Sentiment	Aspect Category
1	Why is then Amazon took up the sacred duty of selling such an item which they can't exchange/ have no control ? The product developed new issues like proximity sensor malfunction and last but most importantly loosing mobile network every other minute(even had two software updates)	Negative	Software
1	May be my first negative review about the product & Amazon both	Negative	General
1	May be my first negative review about the product & Amazon both	Negative	Amazon

As of now, interpretation of emoticons by the users has not been taken into consideration and emoticons have been removed when found. The dataset has been divided into 70:30 for training and testing. Table 5.4 displays the statistics for the scraped data. It lists the predefined aspect categories that were identified prior to the start of the manual tagging process.

Table 5.4: Dataset Statistics

Aspect Categories	Polarity			
	Positive	Neutral	Negative	Total
Accessory	14	4	49	67
Amazon (Service +Seller)	28	8	32	68
Battery	147	9	38	194
Brand	103	10	20	133
Camera	221	11	39	271
Delivery	47	2	17	66
Display	72	9	40	121
General	491	47	45	583
Hardware	32	7	40	79
Mobile_Looks	64	6	21	91
OS	57	5	17	79
Performance	45	0	24	69
Price	108	10	39	157
Processor	6	3	6	15
Software	50	4	37	91

5.3.4 Manual ABSA-annotation steps

Six annotators have identified the aspect category from a predefined list of aspect categories and expressed their polarity (positive, neutral, or negative) towards the identified aspect. A total of 15 potential aspect categories of a mobile phone are identified. For tagging, the dataset is divided equally among three annotators. In the event of a tagging conflict, the authors have made the final decision. Table 5.5 displays information about the annotators of the dataset.

Table 5.5: Dataset Annotators Details

Annotator ID	**Profession**	**Task**
1	Research Scholar	Data Collection and final annotation
2	Faculty, Author	Final decision on annotation
3	Faculty, Author	Final decision on annotation
4	Post Graduate Student	Initial annotation of dataset
5	Post Graduate Student	Initial annotation of dataset
6	Post Graduate Student	Initial annotation of dataset

5.3.5 Manual ABSA-baseline experiments and results

Based on supervised machine learning, models such as Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LG), Random Forest (RF), K Nearest Neighbor (KNN), and Deep Learning Model (Keras-MLP) using Keras Sequential Model API in Python are constructed to identify the best model for classifying the reviews. Aspects such as memory, mobile quality, dust resistance, and water resistance are discarded in the machine learning process due to significantly fewer reviews. By removing them from the training and testing datasets, the accuracy of all models has improved significantly, depicting the application of annotated dataset, namely the detection of aspect category and sentiment.

Accuracy (ACC) is defined as the proportion of correctly classified reviews divided by the total number of reviews, as defined in Equation 5.19. It is a widely used metric for assessing the performance of classification methods (Kou et al., 2020). Higher accuracy is preferred.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.19}$$

where, TP stands for True Positive, i.e. percentage of actually correctly classified reviews that are predicted classified correctly FP stands for False Positive, i.e. percentage of incorrectly classified reviews that are predicted correctly. FN stands for False Negative i.e. percentage of incorrectly classified reviews that are predicted incorrectly. TN stands for True Negative i.e. percentage of correctly classified reviews that are predicted incorrectly.

Table 5.6 shows the accuracy scores from the training and testing phases for classifying review sentences based on their aspect categories and sentiments. The best results are highlighted in bold and green, while the worst results are highlighted in bold and red. The result is obtained through 3-fold cross-validation, and hyperparameter tuning is used to improve the accuracy of these models.

Table 5.6: Accuracy of Machine Learning Models on Proposed Dataset

	Accuracy -Aspect Category		**Accuracy- Aspect Sentiment**	
Model	**Training Phase**	**Testing Phase**	**Training Phase**	**Testing Phase**
Keras-MLP	0.8896	0.6745	0.977	0.763
LR	0.9241	0.6319	0.9864	0.7709
KNN	0.5325	0.4992	0.8679	0.7725
NB	0.878	0.5719	0.981	0.793
RF	0.7168	0.6319	0.9018	0.7757
SVM	0.8656	0.6398	0.8875	0.7946

The precision, recall and f-measure scores obtained are calculated through Equation 5.20, 5.21 and 5.22 respectively as:

$$Precision(P) = \frac{TP}{TP + FP} \tag{5.20}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{5.21}$$

F-measure combines both precision and recall as follows:

$$f - measure(f) = 2 * \frac{P * R}{P + R} \tag{5.22}$$

Table 5.7 shows the precision (P), recall (R) and f-measure (F1) for train and test data for the built models.

Table 5.7: Evaluation of Machine Learning Models on Proposed Dataset

Model	Testing Phase			Testing Phase		
	Aspect Category			Aspect Sentiment		
	Precision	Recall	F-Score	Precision	Recall	F-Score
Keras-MLP	0.66	0.63	0.71	0.75	0.76	0.77
KNN	0.56	0.5	0.45	0.76	0.77	0.75
LR	0.63	0.63	0.62	0.74	0.77	0.75
NB	0.56	0.57	0.56	0.75	0.79	0.77
RF	0.63	0.63	0.6	0.75	0.78	0.73
SVM	0.62	0.64	0.62	0.74	0.79	0.75

As shown in Figure 5.9, the deep learning model (MLP model built using Keras API) for classifying review text into fifteen predefined aspect categories produces the most accurate result in the testing phase, with an accuracy of 67.45%. K- nearest neighbor performs the worst in this task, achieving only 49.92% accuracy.

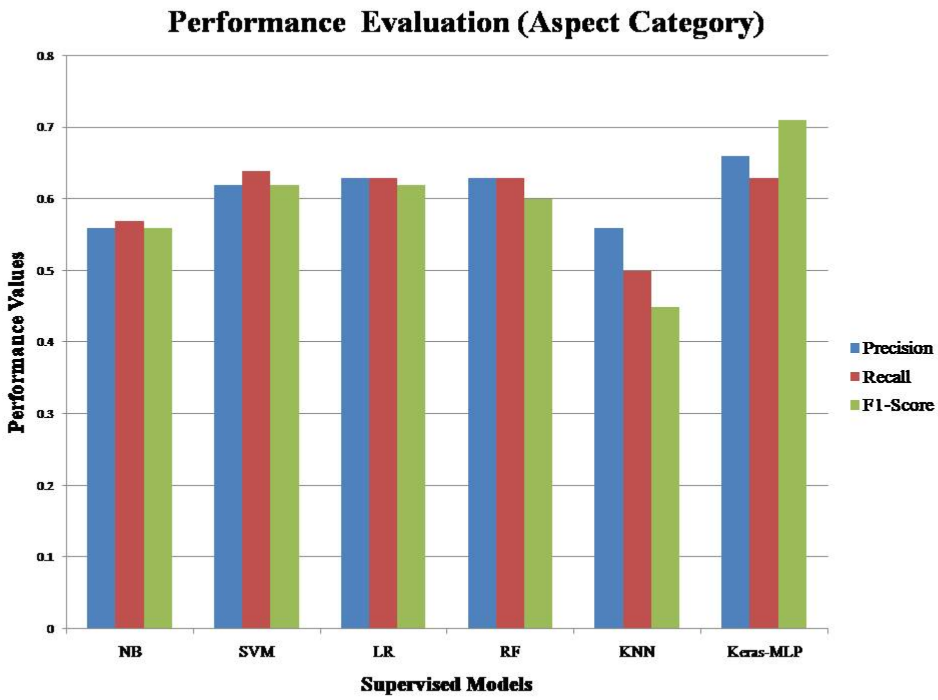

Figure 5.9: Performance of Machine Learning Models on Proposed Dataset (Aspect Category)

This experimental evaluation is conducted through the Keras module for deep learning and the scikit-learn module for machine learning provided by open-source

Python software (Pedregosa et al., 2011; Chollet, 2015; Mueller, 2020). Standard Windows system with 64 bit Intel Core i3 CPU @2.00 GHz, 2000 MHz, 2 Core(s), 4 logical processors and 4.00 GB RAM is used for training and testing the ABSA system.

As shown in Figure 5.10, with an accuracy of 79.46%, Support Vector Machine is the most accurate for classifying review text into three predefined aspect sentiments. The MLP model has the lowest accuracy with 76.30% for aspect sentiment classification.

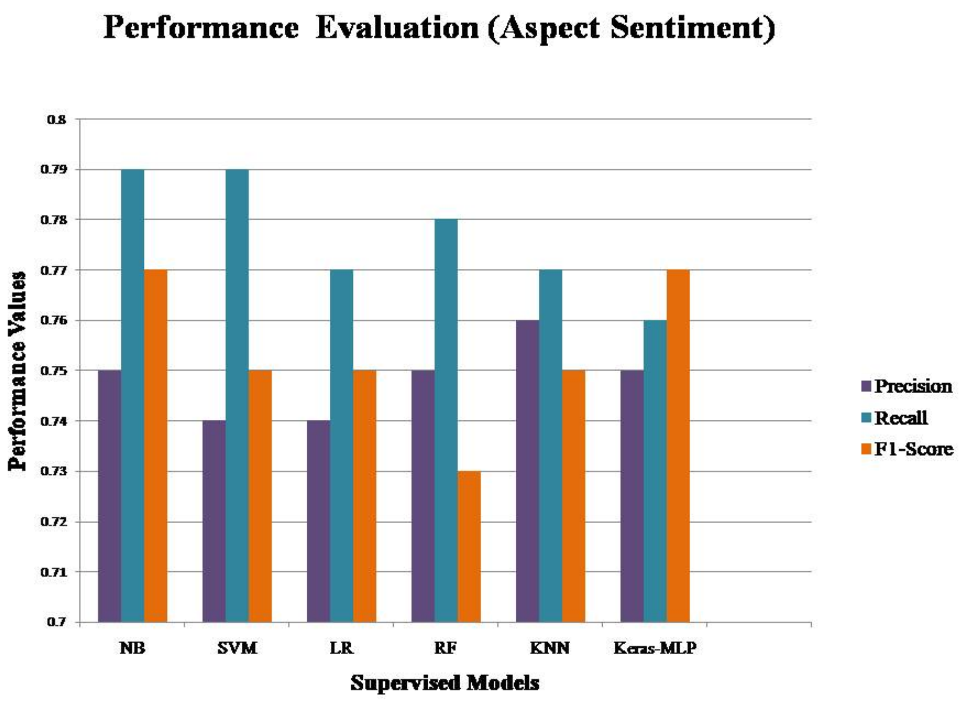

Figure 5.10: Performance of Machine Learning Models on Proposed Dataset (Aspect Sentiment)

For each of the above models standard architecture commonly followed is initially used. This architecture is then optimized for performance by tuning (trying out all possible combinations of hyperparameters to achieve the best possible output) the hyperparameters using the GridSearchCV method under the scikit-learn library in Python.

The hyperparameters tuned for the stated machine learning models for aspect category classification are shown in Table 5.8.

Table 5.8: Hyperparameters Tuned for Machine Learning Models for Aspect Category Classification

NB	**SVM**	**LR**	**RF**	**KNN**	**Keras-MLP**
alpha =0.01	alpha =0.001	C =100000.0	ccp_alpha =0.0	algorithm ='auto'	Batch_size =100
class _prior =None	average =False	dual =False	bootstrap =False	leaf_size =30	Epochs =50
fit_prior =True	class_weight =None	class_weight =None	class_weight =None	metric ='minkowski'	
	early_stopping =False	intercept _scaling =1	criterion ='gini'	metric_params =None	
	epsilon =0.1	multi_class ='auto'	max_depth =50	n_jobs =None	
	eta0 =0.0	random_state =None	max_features ='sqrt'	n_neighbors =11	
	fit_intercept =True	fit_intercept =True	max _leaf_nodes =None	p=2	
	l1_ratio =0.15	l1_ratio =None	max_samples =None	weights ='uni-form'	
	learning _rate =op-ti-mal	solver ='saga'	min_impurity _decrease =0.0		

Continued on next page

Table 5.8 – continued from previous page

NB	**SVM**	**LR**	**RF**	**KNN**	**Keras-MLP**
	loss ='hinge'	max_iter =100	min_impurity _split =None		
	max_iter =100	n_jobs =None	min_samples _leaf =2		
	n_iter_no _change =5	penalty ='none'	min_samples _split =10		
	n_jobs =None	tol =0.0001	min_weight _fraction_leaf =0.0	algorithm ='auto'	
	penalty ='l2'	verbose =0	n_estimators =800	algorithm ='auto'	
	power_t =0.5	warm_start =False	n_jobs =None		
	random_state =None		oob_score =False		
	shuffle =True		verbose =0		
	tol=0.001		warm_start =False		
	validation _fraction =0.1				
	verbose =0				
	warm_start =False				

The hyperparameters tuned for the stated machine learning models for aspect sentiment classification are shown in Table 5.9 below.

Table 5.9: Hyperparameters Tuned for Machine Learning Models for Aspect Sentiment Classification

NB	**SVM**	**LR**	**RF**	**KNN**	**Keras-MLP**
alpha =0.01	alpha =0.001	C =100000.0	ccp_alpha =0.0	algorithm ='auto'	Batch_size =500
class _prior =None	average =False	dual =False	bootstrap =False	leaf_size =30	Epochs =50
fit_prior =True	class_weight =None	class_weight =None	class_weight =None	metric ='minkowski'	
	early_stopping =False	intercept _scaling =1	criterion ='gini'	metric_params =None	
	epsilon =0.1	multi_class ='auto'	max_depth =50	n_jobs =None	
	eta0 =0.0	random_state =None	max_features ='sqrt'	n_neighbors =11	
	fit_intercept =True	fit_intercept =True	max _leaf_nodes =None	p=2	
	l1_ratio =0.15	l1_ratio =None	max_samples =None	weights ='uniform'	
	learning _rate =optimal	solver ='saga'	min_impurity _decrease =0.0		
	loss ='hinge'	max_iter =100	min_impurity _split =None		

Continued on next page

Table 5.9 – continued from previous page

NB	SVM	LR	RF	KNN	Keras-MLP
	max_iter =1000	n_jobs =None	min_samples _leaf =2		
	n_iter_no _change =5	penalty ='none'	min_samples _split =10		
	n_jobs =None	tol =0.0001	min_weight _fraction_leaf =0.0		
	penalty ='l2'	verbose =0	n_estimators =800		
	power_t =0.5	warm_start =False	n_jobs =None		
	random_state =None		oob_score =False		
	shuffle =True		verbose =0		
	tol=0.001		warm_start =False		
	validation _fraction =0.1				
	verbose =0				
	warm_start =False				

The input features are limited to a maximum of 2000 words and are fed into the sequential model via the input_dim parameter. The chosen model is made up of two dense layers: the first layer is made up of 512 neurons, and the second layer is made up of 15 neurons, the output of which is mapped to 15 categories in the case of aspect category classification and three neurons in the case of aspect sentiment classification.

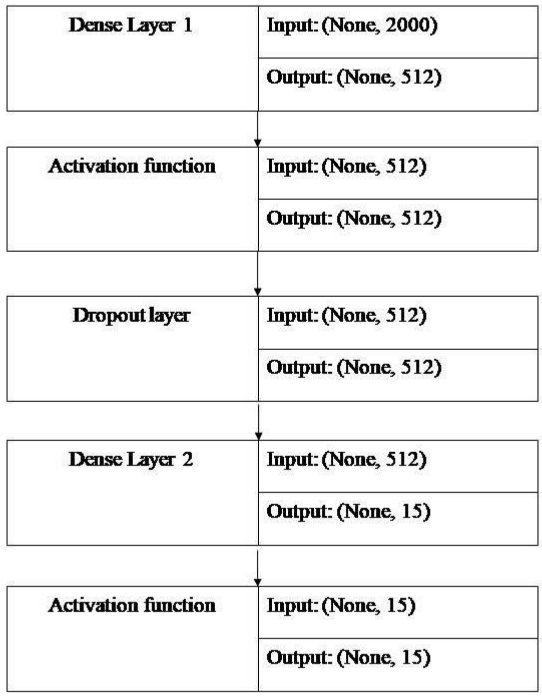

Figure 5.11: Architecture of Sequential Model

Figure 5.11 depicts the architecture of the sequential model. *ReLu* activation function is used in the first layer and *softmax* activation function in the output layer. A *dropout* of 0.5 is added to set the fraction of inputs to zero to reduce overfitting. The model is compiled using *categorical cross-entropy* loss and optimized with the stochastic gradient descent method known as the *adam* optimizer. A validation split of 10% is configured while fitting the model to check the model for training and validation accuracies over all the epochs. Table 5.10 is supplementary table that details the performance of the Sequential model for aspect category classification.

Table 5.10: Performance Evaluation of Deep Learning Model (Aspect Category)

Train epochs=50 & batch_size=100					**Validation epochs=50 & batch_size=100**				
Loss	**Acc**	**F1_Score**	**P**	**R**	**Loss**	**Acc**	**F1_Score**	**P**	**R**
0.268	0.890	0.946	0.859	0.898	1.298	0.675	0.710	0.627	0.665

While the validation loss is 1.298, accuracy is 0.675, f1_score is 0.710, precision is 0.627 and recall is 0.665, the training loss is 0.268, accuracy is 0.890, f1_score is 0.946, precision is 0.859 and recall is 0.898 for aspect category classification.

Table 5.11 is supplementary table that details the performance of the Sequential model for aspect sentiment classification.

Table 5.11: Performance Evaluation of Deep Learning Model (Aspect Sentiment)

Train epochs=50 & batch_size=500					**Validation epochs=50 & batch_size=500**				
Loss	**Acc**	**F1_Score**	**P**	**R**	**Loss**	**Acc**	**F1_Score**	**P**	**R**
0.080	0.977	0.978	0.977	0.977	0.827	0.763	0.768	0.747	0.757

While the validation loss is 0.827, accuracy is 0.763, f1_score is 0.768, precision is 0.747 and recall is 0.757, the training loss is 0.080, accuracy is 0.977, f1_score is 0.978, precision is 0.977 and recall is 0.977 for aspect sentiment classification. Figure 5.12 represents the model accuracy and model loss of the sequential model for aspect category classification.

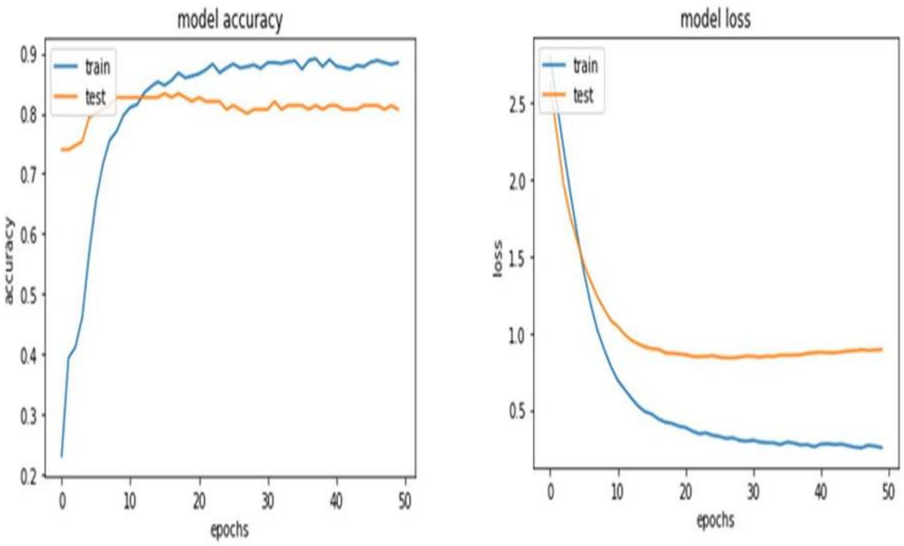

Figure 5.12: Sequential Model Accuracy and Loss Corresponding to Aspect Category Classification

As shown in Figure 5.12, the sequential model accuracy for testing phase for aspect sentiment classification is almost constant after 30 epochs and is about a point less than the training phase.The model loss testing phase for aspect sentiment classification is constant after 20 epochs.

Figure 5.13 represents the model accuracy and model loss of the sequential model for aspect sentiment classification.

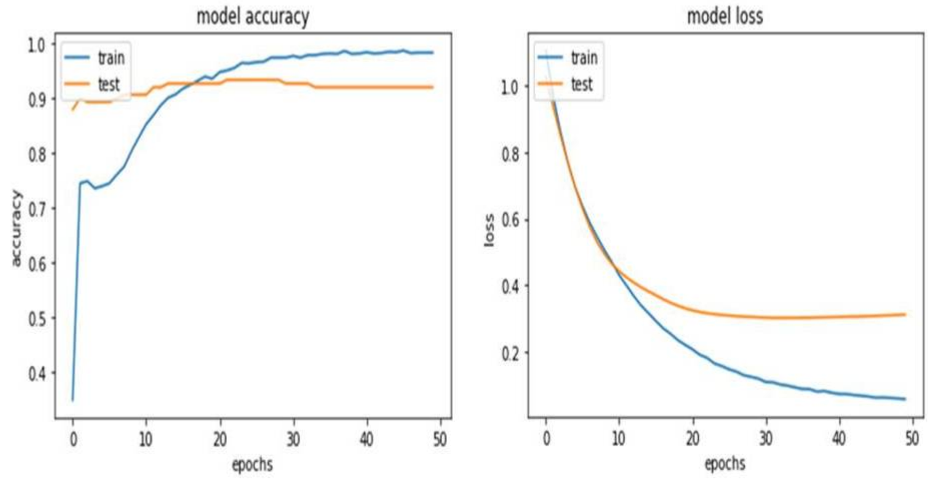

Figure 5.13: Sequential Model Accuracy and Loss Corresponding to Aspect Sentiment Classification

5.3.6 Manual ABSA-limitation

A dataset for ABSA of mobile phone reviews has been provided. The dataset has been designed to automate aspect category extraction and aspect category polarity identification using machine learning techniques. Furthermore, the constructed dataset has been evaluated using several state-of-the-art machine learning techniques. Understanding the intent conveyed by emoticons has not been taken into account in this study. In addition, the abbreviated words have not been addressed. The dataset collected for a single entity – Apple-iPhone11 mobile Phone – has less than 1000 reviews, resulting in a small corpus of the labeled dataset but with significant results. The above mentioned satisfactory results are generated using actual imbalanced data, which can be improved by balancing the dataset. The MLP sequential model is the most accurate when the number of predefined aspect categories are fifteen and the least accurate when the number of predefined aspect sentiments are three, indicating a need for more data for the training process. Traditional ML model Support Vector Machine performs the best when only three predefined aspect sentiments are to be classified. The majority of the ML models achieve satisfactory accuracies ranging from 49 to 67% for aspect category classification and 76 to 79% for aspect sentiment classification. As a result, this dataset of mobile phone reviews in English can serve as a benchmark for ABSA.

5.3.7 Automatic ABSA

The complete design of the proposed approach to mitigate sparsity is depicted in Figure 5.14.

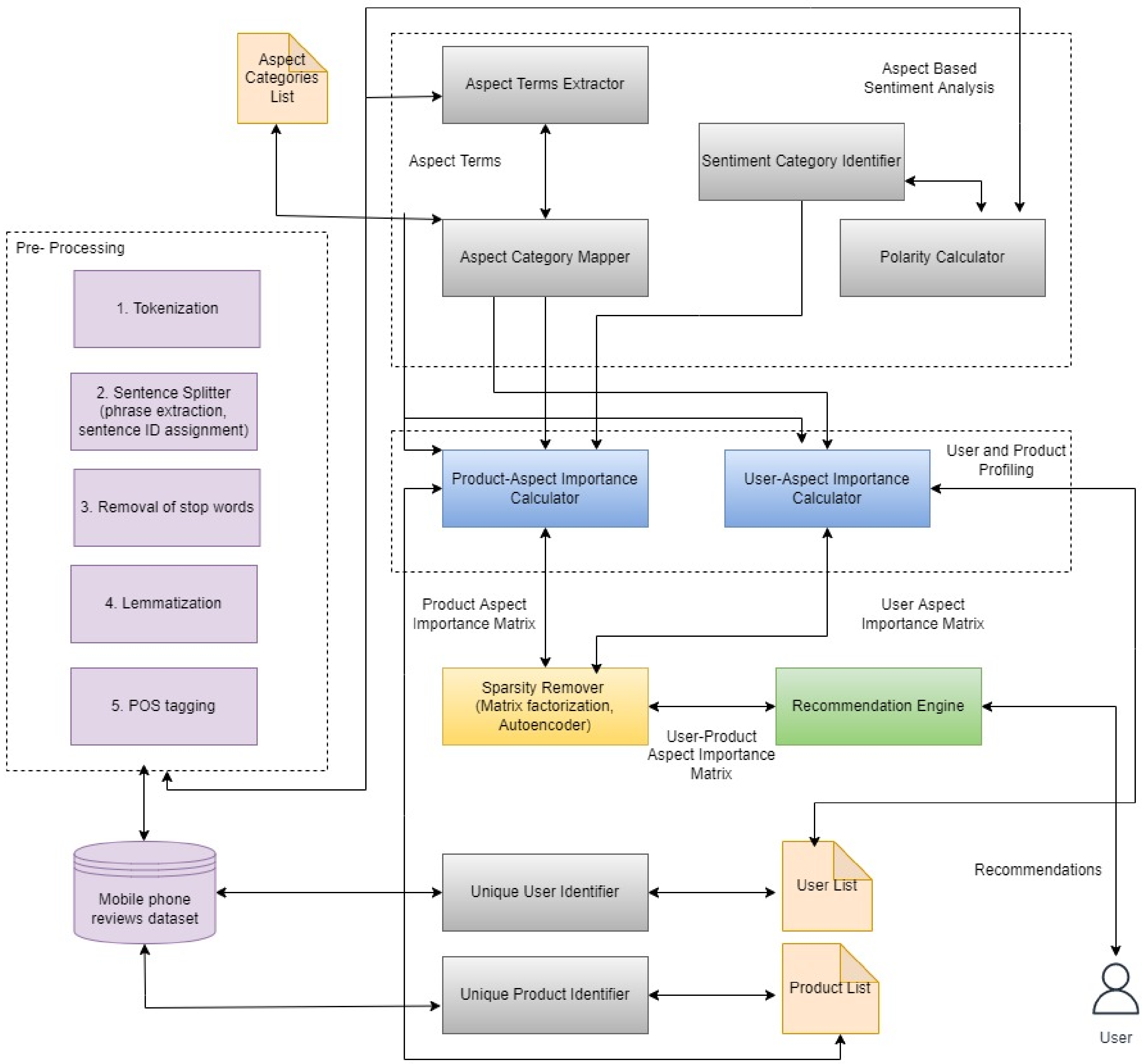

Figure 5.14: Design of the Proposed Approach

Manual ABSA requires lot of human efforts to annotate the aspect sentiment and category. Such kind of tagging is not possible with large datasets. Also, the above dataset caters to a single mobile phone entity. The recommendation algorithms can not function with only one item information. This led to the conceptualisation of Automatic ABSA. Automatic ABSA is followed by construction of user and product profile based on the frequency of occurrence of aspects and their corresponding sentiments (Zhang et al., 2020). The matrices obtained are checked for sparsity levels and sparsity is removed through matrix factorization and autoencoder before being fed to baseline recommendation algorithms.

The steps included in ABSA are identification of aspect terms from review sentences, extraction of polarity for aspect terms and detection of aspect categories from aspect terms. The detail of these steps are as follows:

5.3.8 Automatic ABSA-identification of aspect terms from review sentences

The aspect terms for mobile phone review dataset commonly consist of phone, volume, money, battery, keyboard, screen, service, cost, purchase etc. These terms have been identified as noun terms and have been extracted through Part-of-Speech (POS) tagging using nltk library of Python programming language. For instance, the following aspect terms were extracted for the review sentence “is a good product and I is working very well, this provider met my expectations by giving me a good quality product, I am very pleased with this purchasethanks”, product, provider, expectation quality product and purchasethank.

5.3.9 Automatic ABSA-extraction of polarity for aspect terms

A review sentence may provide positive, neutral or negative feedback about one or more aspect terms. For easy and quick determination of polarity of aspect terms extracted in the above step, the review sentences were split initially using the full stop and then using comma as the sentence splitter in order to obtain phrases of sentences. Each phrase of the same review sentence was assigned the same review ID. The polarity of each phrase was determined using vaderSentiment, a popular library in Python for sentiment analysis. With the help of sentiment lexicon, a review sentence is considered to be positive if the outputted compound score is larger than 0.05, for a score smaller than -0.05, it is considered as negative and neutral for the remaining values.

For instance, for the review sentence mentioned in the previous step, the sentiments identified for the phrases of the sentence are stated in Table 5.12 below.

Table 5.12: Example: Aspect Term, Category, Sentiment Score and Polarity Identification

S.No	Sentence Phrase	Aspect Terms	Aspect Category	Sentiment Score	Polarity
1	Pretty good for a used phone	phone	Phone	0.727	Positive
2	Babied the hell out of the phone btw so imagine my suprise	phone suprise	Phone	-0.6808	Negative
3	good condition	condition	Performance	0.4404	Positive
4	['is a good product', 'I is working very well', ' this provider met my expectations by giving me a good quality product', ' I am very pleased with this purchasethanks']	['product', 'NA', 'provider expectation quality product', 'purchasethank']	['Battery', 'Battery', 'Quality', 'NA']	[0.4404, 0.3384, 0.6486, 0.4927]	['Positive', 'Positive', 'Positive', 'Positive']

5.3.10 Automatic ABSA-detection of aspect categories from aspect terms

Based on the authors' product and Amazon platform's domain knowledge, ten predefined aspect categories were identified as Battery, Camera, Delivery, Display, Performance, Price, Phone, Quality, Sound, Size. A mapping step was followed to map the aspect terms extracted. to these pre-defined aspect categories. Lexical database for English language, Wordnet was utilised to identify synonyms of the identified aspect terms. The mappings were done based on the values generated for the semantic relatedness of the aspect terms to the aspect categories. The aspect term was assigned to the aspect category with maximum semantic relatedness value.

The annotation done above shall help in prediction of aspect category and aspect sentiment after suitable training of supervised machine learning algorithms. Such algorithms are dependent on the availability of labeled dataset for making these predictions.

5.3.11 User and product profiling

For the current presentation, user reviews have emerged as an important side information source to gauge user(s) preferences.

Aspect Importance for User (AIU): The importance of aspect a_i for user u is given through Equation 5.23 as:

$$Imp(u, a_i) = \frac{f_{u,a_i}}{f_u} * \frac{R}{1 + e^{-f_u}} \tag{5.23}$$

where, f_{u,a_i} is the frequency of aspect a_i in the reviews by user u and f_u is the number of review splits for user. Logistic function $\frac{1}{1+e^{-f_u}}$ has been used to regularize f_u for accurate depiction of user preference (Hou, Yang, Wu, & Yu, 2019). The obtained preference is scaled into rating range value of 1 to 5 by multiplication with R, that is, maximum value on rating scale (5) of a product.

Aspect Importance w.r.t Product (AIP): Product reviews from different users for the same item consists of users' preference for different aspects of that item. Importance of an aspect for an item can be inferred through the sentiment expressed by users. The importance of aspect a_i for product p is given through Equation 5.24 by:

$$Imp(p, a_i) = \frac{f_{u,p_i}}{f_p} * \frac{1}{1 + e^{-f_p}} * S_{a_i} \tag{5.24}$$

where, f_{p,a_i} is the frequency of aspect a_i in the reviews for product p and f_p is the number of reviews for a product. Logistic function $\frac{1}{1+e^{-f_p}}$ has been used to regularize f_p for accurate depiction of product preference and S_{a_i} represents the sentiment score of the aspect.

5.3.12 Sparsity removal

The algorithm of the proposed method for mitigation of this type of sparsity is as follows:

Input: User ID, Product ID, Product name, Review Text, Rating Output: Recommendation list for each user

1. *Rec_List_u=[],$Aspect\_Term_s$=[], $Aspect\_Category_s$=[], Sentence_phrase=[], Phrase_polarity=[], i=10, Resultant_Matrix=[], Non-Resultant_Matrix=[]*

2. *select 10000 random review sentences from the chosen dataset*

3. *for each review 'r'*

   (a) *tokenize 'r' to generate tokens*

(b) *split 'r' into sentence phrases 'p' with occurrence of keywords and, or, but, plus, by the way, also and punctuation symbol representing comma, exclamation mark and full stop and assign common review ID to all the sentence phrases of a review*

(c) *for each sentence phrase 's'*

   i. *remove stop words*

   ii. *remove words with word length less than 3*

   iii. *lemmatize the tokens*

   iv. *perform part-of-speech tagging*

   v. $Aspect\_Term_s$ *= nouns extracted from POS tagging in step iv. above*

   vi. *for each aspect term in* $Aspect\_Term_s$

      A. $Aspect\_Category_at$ *= map each aspect term to predefined aspect categories using wordnet*

   vii. *end for*

(d) *identify Phrase_polarity using vaderSentiment library*

(e) *end for*

4. *end for*

5. *for each user 'u'*

   (a) *find aspect importance to user, Imp(u,*a_i*)*

6. *end for*

7. *for each product 'p'*

   (a) *find aspect importance to user, Imp(u,*a_i*)*

8. *end for*

9. *Obtain matrix for aspect importance for user (AIU) and aspect importance for product (AIP)*

10. *Calculate sparsity of matrices AIU and AIP obtained in Step 9*

11. *If sparsity is greater than 50% then*

    (a) *Remove sparsity using matrix factorization or auto encoder*

    (b) *Normalize the matrices value to an interval range of 1 to 5*

    (c) *Resultant_Matrix = Multiply AIU and AIP*

(d) *Stack Resultant_Matrix as triplet of (User, Product, Aspect Weight)*

(e) *Input Resultant_Matrix to Baseline CF techniques to obtain Rec_List*

12. *else*

(a) *NonResultant_Matrix= Multiply AIU and AIP*

(b) *Stack NonResultant_Matrix as triplet of (User, Product, Aspect Weight)*

(c) *Input NonResultant_Matrix to Baseline CF techniques to obtain Rec_List*

13. *End if*

To evaluate the recommendation performance, several state-of-the art recommendation models including SVD, SlopeOne, Centered KNN, CoCluster, Baseline, Random Predictor have been implemented using Python's surprise library (Hug, 2020). The initial review information has been processed to obtain user, product and aspect weight importance to serve as input for these algorithms. The obtained non-sparse input through matrix factorization and autoencoder was split into training and testing dataset in the ratio of 70:30. The recommendation algorithms have been trained using the train set and tested using a test set to measure RMSE, MAE and MSE (Da'u et al., 2020).

5.4 RESULT & DISCUSSION

Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Squared Error (MSE) have been used in the literature and in this study to assess the rating prediction of recommendation systems.

Table 5.13: Performance of State-of-Art Recommendation Algorithms

S.No	Technique	Input Type	RMSE	MAE	MSE
1	SVD	Sparse	1.4001	1.1530	1.9603
		Non-Sparse (MF)	0.0246	0.0127	0.0006
		Non-Sparse (AE)	0.8340	**0.6953**	0.6956
2	SlopeOne	Sparse	1.6530	0.8993	2.7325
		Non-Sparse (MF)	0.0406	**0.0245**	0.0016
		Non-Sparse (AE)	0.8344	**0.6956**	0.6962
3	Centered KNN	Sparse	1.5242	0.5911	2.3232
		Non-Sparse (MF)	0.0045	**0.0018**	1.984e-05
		Non-Sparse (AE)	0.8339	**0.6971**	0.6955
4	CoClustering	Sparse	1.6384	0.9091	2.6843
		Non-Sparse (MF)	0.3859	**0.3488**	0.1489
		Non-Sparse (AE)	0.8345	**0.6981**	0.6963
5	Baseline Predictor	Sparse	1.6547	0.9033	2.7382
		Non-Sparse (MF)	0.0284	**0.0147**	0.0008
		Non-Sparse (AE)	0.8339	**0.6953**	0.6954
6	Random Predictor	Sparse	2.9663	2.2400	8.7990
		Non-Sparse (MF)	0.3065	**0.2329**	0.0939
		Non-Sparse (AE)	0.8470	**0.7078**	0.7174

As shown in Table 5.13, all of the recommendation algorithms outperform in terms of the three evaluation metrics when inputted with non-sparse values as per the proposed method. The result justifies and implies good efficiency of the proposed aspect based approach on the state-of-the art recommendation algorithms. The non-sparse output obtained through the matrix factorization technique yields the least error values for Centered KNN method and highest error values for CoCluster method. But, the non-sparse output generated through the autoencoder method yields the least error values for the Baseline Predictor method and highest error values for the Random Predictor method. The best and worst error values have been highlighted in green and red respectively. Also, although autoencoder is a deep neural network technique and its error values are more than the mathematical based matrix factorization technique, but the time taken by autoencoder training for generating the sparse values is 83.95 seconds which is very less than time taken by matrix factorization technique, which is around 25 minutes. Hence, autoencoder technique fares much better than matrix factorization technique. Further, the obtained error values in the worst cases from both the techniques are far lower than the case when sparse input is provided to the same algorithm, again justifying the proposed approach in this study.

5.5 CHAPTER SUMMARY

In this chapter, user reviews which form one of the important sources of side information have been leveraged to improve the performance of recommender system. A new kind of sparsity problem originating due to subjectivity of reviews has been explained and alleviated with the help of matrix factorization and autoencoder technique. Manual aspect-based sentiment analysis, automatic aspect-based sentiment analysis, user and product profiling, sparsity removal through matrix factorization and autoencoder have been performed. As part of manual ABSA, a dataset has been contributed to perform ABSA of mobile phone reviews in English language. Such a dataset does not exist in the literature devoiding researchers' analysis of user reviews and their corresponding classification. The dataset has also been validated for performance through several machine learning algorithms. MLP sequential model performed the best for aspect category classification and SVM performed the best for aspect sentiment classification.

Owing to the limitations of manual ABSA, automatic ABSA has been performed, followed by profiling of both the users and products so as to capture their inclination towards product's aspects. Sparsity arising due to subjectivity of reviews has been successfully mitigated through matrix factorization and autoencoder. State-of-the-art recommendation system algorithms have then been compared and contrasted for performance with sparse and non-sparse input matrices. The experiments conducted reveal that performance of the recommender system after removal of sparsity is the best and improves by high margins. RMSE, MAE, MSE of collaborative filtering recommendation algorithms with non-sparse input show significant improvements thereby justifying the proposed approach in this study.

CHAPTER 6

IDENTIFICATION OF USEFUL REVIEWS THROUGH MACHINE LEARNING

CHAPTER VI

IDENTIFICATION OF USEFUL REVIEWS THROUGH MACHINE LEARNING

This chapter identifies useful product reviews through machine learning algorithms namely, Logistic Regression, Decision Tree, Random Forest, Ada Boost, Gradient Boost, Extra Trees, K Nearest Neighbor and Linear Discriminant Analysis. In addition to features of the chosen dataset, additional features have been derived through feature engineering. Pre-processing, feature engineering, model training, model testing on the original and derived features is performed and then the machine learning models are evaluated for accuracy, area under the curve, precision, recall, f1 score, Kappa score and Mathew correlation coefficient metrics.

6.1 INTRODUCTION

For e-commerce users and its stakeholders, online customer reviews have grown into electronic word of mouth (eWoM) (Saumya, Singh, Baabdullah, Rana, & Dwivedi, 2018; Saumya, Singh, & Dwivedi, 2020). These customer reviews contain elaborated experiences of customer with the products. They aid consumers in making purchase decisions and highlight any necessary quality improvements required, hence assisting commercial organisations in increasing the product sales. This entails analysing these customer reviews (Du et al., 2020). To extract the customers' preference towards a product, sentiment analysis or topic modeling approaches are utilised. This assists in constructing the customer profile and determining the customer's preferences for unseen products. Many platforms, including Amazon, Yelp, TripAdvisor, IMDB, and Netflix, host large numbers of user reviews. Unfortunately, the ever-increasing quantity of items, customers, and product reviews on the e-commerce platform has resulted in the problem of information overload making it impossible for customers to read all the product reviews. To address this issue, the ability for other customers to mark a review as helpful had been introduced. A product's star rating reflects a user's experience with a product but a review's number of votes indicates its usefulness. But, due to factors such as the immense volume of electronic word of mouth, the voluntary helpfulness voting method, the level of visibility, and the reviews' recency, not all reviews acquire this helpful vote (Arif et al., 2019; Mauro et al., 2021). The solution to information overload consists of browsing user reviews based on their helpfulness or utility (Ge et al., 2019). Thus, the purpose of this study is to classify the product review according to its usefulness. This will not only assist buyers determine whether a product is beneficial or not, even if the review has not received any votes, but it may also be fed into the recommender system in order to generate useful recommendations for users.

Eight distinct machine learning models, including Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), AdaBoost (ADA), Gradient Boost (GB), Extra Trees (ET), K Nearest Neighbors (KNN), and Linear Discriminant Analysis (LDA), have been trained and tested on existing and derived features and evaluated on seven evaluation metrics, including Area under the Curve (AUC), Accuracy (ACC), F1-score (F1), Precision (P). The best model has been fine tuned to predict the usefulness of reviews. Review factors such as overall rating, user review, review summary, review votes, word count of review, character count of review, review's sentiment score, and average word count of review have been utilised to determine the review's usefulness. In addition to the features currently included in the selected dataset, such as the overall rating, user review, review summary, and review votes, additional features extracted from user reviews have been employed as input to the prediction model. This study will help customers to identify valuable reviews and enable e-commerce managers, merchants, and retailers to optimise the listing of product reviews based on the usefulness of the reviews.

6.2 DATASET DESCRIPTION

Amazon cell phone and accessories dataset has been considered for predict useful reviews (McAuley, 2018; Ni, Li, & McAuley, 2020). The selected dataset consists of 1048572 rows and 12 columns namely reviewerID, asin, reviewerName, vote, style, reviewText, overall, summary, unixReviewTime, reviewTime and image. While column vote, overall are numeric columns, rest of the columns are alphanumeric in nature. The description of the columns of the selected dataset is shown in Table 6.1:

Table 6.1: Dataset Description

Column name	Column description
reviewerID	ID of the reviewer, e.g. A284QS51P9P9V1
asin	ID of the product, e.g. B00UVSNVHA
reviewerName	name of the reviewer
vote	helpful votes of the review
style	a dictionary of the product metadata, e.g., ”Format” is ”Hardcover”
reviewText	text of the review
overall	rating of the product
summary	summary of the review
unixReviewTime	time of the review (unix time)
reviewTime	time of the review (raw)
image	images that users post after they have received the product

6.3 METHODOLOGY

This section presents the methodology used to identify useful reviews. The steps undertaken as part of prediction of useful reviews are shown in Figure 6.1:

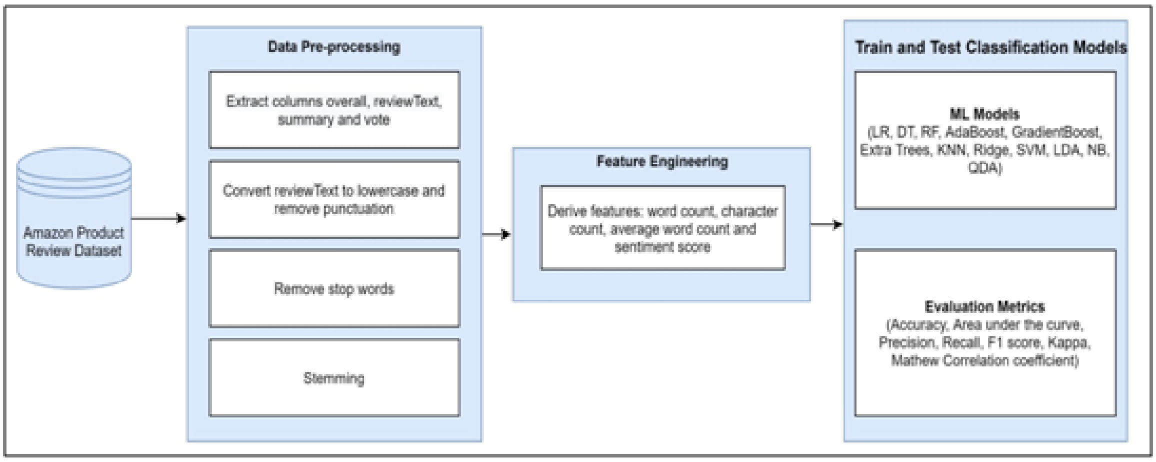

Figure 6.1: Methodology to Identify Useful Reviews

Incorporating reviews' usefulness score helps to solve the problem of browsing of a large number of reviews in order to arrive at a purchase decision. As all reviews are not tagged with helpfulness scores, it is essential to predict the useful reviews (Ge et al., 2019). The steps undertaken as part of prediction of useful reviews are as follows:

6.3.1 Pre-processing

When adopting machine learning models to predict an outcome, it is necessary to provide clean data. Models rely heavily on prepared text in order to be successful. In order to extract the main knowledge from the text data, which is unorganised, additional processing is required.In order to categorize the reviews as useful or useless and clean the input data, the following pre-processing steps have been undertaken:

1. Out of the 12 columns available, only columns- overall (represents product rating), reviewText, summary and vote have been utilized.

2. ReviewText column has been converted to lowercase and punctuation has been removed. Conversion to lowercase is the procedure of changing the review text's complete words to lowercase letters (Alsubari, Deshmukh, Al-Adhaileh, Alsaade, & Aldhyani, 2021).

3. After performing the below mentioned feature engineering steps, stop words such as 'a', 'an', 'the' using Python's nltk library have been removed (Kabir, Kabir, Xu, & Badhon, 2019). Stop words are terms that are used very frequently yet do not contribute significantly to the meaning of any analysis. The removal of stop words brings a reduction in dimensionality. Stop words are grammatical constructions that serve no use in the context of the documents. Examples of stop words include prepositions, determiners, and coordinating conjunctions. Stop words are words that are not considered to have any value as keywords, hence they are removed from texts used in text minimization systems.

4. Step 3 has been followed by a stemming process in which Porter stemmer has been used to apply stemming on the reviewText column. Stemming is a simple processing step that eliminates a word's prefixes, infixes. A words conjugation form is reduced to its root form. For instance, the English word "eliminate" can be modified with its rules to generate the phrase elimination. It derives from the root word "eliminate."

6.3.2 Feature engineering

Feature engineering refers to selection of features apt to be considered as input for further processing. In this study, apart from the features considered during the pre-processing phase, four new features have been derived. These features have been described as follows:

1. *Word count:* This column represents the number of words in a review
2. *Char count:* This column indicates number of characters in a review
3. *Avg word count:* This column stands for average word length of a review
4. *Sentiment score:* This column represents polarity of a review ranging from minus one (indicating extremely negative) to plus one (indicating extremely positive) which has been determined with the help of Python's vaderSentiment library

6.3.3 Preliminary analysis

The top ten most frequently occurring words, as shown in Table 6.2, after removal of stop words from the dataset are given below:

Table 6.2: Top Ten Frequently Occurring Words

Word	**Frequency**
Phone	165691
case	117779
one	62104
screen	57831
like	51122
use	43841
great	39611
battery	39595
would	38616
good	37078

As the dataset is related to cell phones, the top ten frequently occurring words are related to this domain. The users have provided reviews mostly related to phone, case, screen and battery. To obtain these words, the frequency of words in the user reviews is obtained and then the top ten words are extracted.

The ten least frequently occurring words in the dataset, with only single occurence are- Performancebattery, gummybearlike, amazonsunvalleytek, knive, terd, hh, nomy, 4siphone, Loosey, caseseems. The review dataset contains the majority of user reviews with the highest rating of the product, that is, 5, followed by user rating 4. The dataset contains more one-star ratings compared to three-star and two-star ratings.

The percentage of overall rating provided by users is provided in Table 6.3.

Table 6.3: Distribution of User Ratings in the Dataset

Rating	5	4	3	2	1
Count	49894	15243	8094	5509	11942
Percentage	55.02	16.81	16.81	6.08	13.17

Supervised learning algorithms require input and output examples for training the model. In order to predict the review usefulness, the target column has been contributed which identifies each review as useful or not. To help the classification models learn if a review is useful or useless all the reviews with more than 10 votes have been marked as useful else useless.

6.3.4 Machine learning models

Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), AdaBoost (ADA), Gradient Boost (GB), Extra Trees (ET), k Nearest Neighbor (KNN) and Linear Discriminant Analysis (LDA) are used to categorize the usefulness of user reviews (Luo & Xu, 2019). All the models have been implemented in Python using the sklearn library.

Machine learning is a branch of Artificial Intelligence in which a computer system automatically identifies hidden patterns from the inputted data and draws conclusions based on the learnt pattern for the unseen data (Han, 2014). The conclusion drawn is termed as desired output or the target. Data plays a very crucial role in the efficient working of machine learning models. In order to identify the underlying patterns of data, the machine learning models undergo a training phase. The training phase requires data which is large in size and pre-processed as input. Pre-processing of data includes cleaning, extraction, transforming of data relevant to the application in consideration. Feeding of unprocessed data to machine learning algorithms leads to generation of incorrect results as the learning takes place on incorrect data. The performance of the training phase decides the number of times the training takes place. In other words, training of the machine learning algorithms or models takes place over several iterations till the desired training phase performance is achieved. Once the training phase is completed, the model proceeds towards the testing phase. In this phase, based on the learnt patterns in the training phase, predictions on unseen data is done. Unseen data implies separate data that wasn't included in the training phase and hence the system or machine is unaware of this data. Segregation of data for the training and testing phase determines the train-test split ratio. For instance, a split ratio of 60:40 implies 60% of the total data is inputted to the training phase and 40% of the total data is inputted to the testing phase. The selection of split ratio depends on the experimental results. Machine learning is classified into following three categories (Portugal et al., 2018; Luo & Xu, 2019; Kadhim, 2019):

1. **Supervised**: In this type of learning, during the training phase, data fed as input to the machine learning algorithms consists of output labels as well. This learning finds its application in both regression and classification problems. When the target value is a continuous variable, the problem is termed as a regression problem, whereas, when the target value is a discrete variable, the problem is termed as a continuous problem. For instance, determining if a review is useful or not useful is a two-class classification problem and determining how much a review is useful is a regression problem. In this study, classifying a review as useful or not useful has been undertaken.

2. **Unsupervised**: In this type of learning, no output label is provided during the training phase. The data is grouped based on the underlying similarities and dissimilarities between the data. For instance, reviews can be grouped together based on the common product aspect being addressed in them.

3. **Reinforcement**: This type of learning is based on reward and punishment mechanism. That is, if the desired outcome is achieved then a positive feedback or reward is given else a negative feedback or punishment is given.

Eight different models have been implemented to classify a review as useful or not useful. The models used are described as follows (Raza et al., 2019):

1. **Logistic Regression**: In this method, given the input vector, the output class is assigned a probability (Raza et al., 2019; Rodrigues et al., 2020). The Logistic Regression model is based on the logistic function or sigmoid function. An S-shaped curve maps real values to values between 0 and 1. The standard notation for the sigmoid function shown in Equation 6.1 is:

$$\frac{1}{1+e^{-z}} \tag{6.1}$$

where, z is any real number to be transformed between 0 and 1. Logistic regression is a multi-class classification problem that began as a binary classification problem. For an input sample z, the probability of being in the first class is given through Equation 6.2 as:

$$p(x) = \frac{e^{\beta^T.z}}{1+e^{\beta^T.z}} \tag{6.2}$$

where, β is parameter vector and z is the training sample. During training, algorithms such as maximum-likelihood estimation are used to minimize errors in the predicted probability.

2. **Decision Tree**: The decision tree is a method that depicts all possible outcomes and the paths leading to those outcomes as a tree structure. The value of each variable is computed in order to form the tree structure and classify the data. Uncertainty level of an element is measured using the concept of entropy. For probability p_i of a class p, entropy is given through Equation 6.3:

$$E(T) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{6.3}$$

The leaves reflect the class level of the tree (T), while the branches represent a combination of input features. For categorization, the best feature (f) is chosen, and until the minimum value in the tree is reached, data is split recursively. (Kabir et al., 2019). Classes are separated into several branches based on Entropy E and Information Gain G given through Equation 6.4 as:

$$I(T, f) = E(T) - E(T|f) \tag{6.4}$$

3. **Random Forest**: It is a discriminative classifier (Hartmann et al., 2019) based on multiple decision trees. A decision tree is made up of nodes and edges, where nodes represent the value of an attribute and edges represent the result of a test (Imtiaz & Islam, 2020). Splitting of the node is based on selection of the best feature. For classification, the test is started at the root node and the edges are followed based on the results; the process is repeated until the leaf node is reached, and finally, the outcome corresponding to the leaf is predicted. Given x input variables, selection of y variables is done so that $m < x$. The predictions are averaged to compute the final prediction value (Mohammed & Kora, 2023).

4. **AdaBoost**: Boosting is an effective approach of ensemble learning through which weak learners are transformed into strong ones by adding weights. These weaker components have better performance due to reduced variance (Shaheen, 2019). AdaBoost or adaptive boosting model starts with placing equal weights to data points initially, followed by placing higher weights to wrongly classified points. This step is continued till minimum error value is reached.

5. **Gradient Boost**: Gradient Boost is a combination of Gradient descent and

boosting technique. The weakness of previous models is identified through gradient. It is more robust to outliers than AdaBoost. It can be used for regression as well as classification problems .

6. **Extra Trees** Extra Trees Classifier stands for Extremely Randomized Trees Classifier. This ensemble learning technique combines the output of several de-correlated decision trees to generate its result. It is similar to the Random Forest classifier but differs in the construction of decision trees. Extra Tree randomly selects cut points to split the nodes. Upon selection of cut points the best feature among the available features subset is selected, thereby adding randomization and optimization. (Shaheen, 2019):

7. **K Nearest Neighbor**: KNN stands for k-nearest neighbor, a statistical classification method. It is a nonparametric classifier from the family of proximity-based algorithms (Kou et al., 2020; Hartmann et al., 2019). In this method, the nearest neighbors of the labeled examples from the training review are ranked for each test review, and then a class assignment is derived using the categories of the highest-ranked neighbors. This model does not learn; instead, it memorizes and represents the entire dataset (Raza et al., 2019). For high dimensional and sparse data, distance computation for the similarity between test and training reviews is computationally expensive.

8. **Linear Discriminant Analysis**: This classifier works by reducing the data to low dimensions and maximizes the separation distance between the target classes. It works on the assumption that the underlying data is linearly separable, has Gaussian distribution and equal covariance matrices of the target classes.

6.3.5 Data setup

Classification estimators are used in this study to predict the user review's usefulness. The target type is binary, with two possible values as useful or useless. The data has been partitioned into 70:30 partitions to obtain the training and testing sets. To allow row shuffling during the train-test split, the data split shuffle is set to true. The predictive models' performance is evaluated using stratified ten-fold cross-validation.

6.4 RESULT ANALYSIS & DISCUSSION

Usefulness is treated as a dependent variable and overall, reviewText, summary, vote, word count, character count, average word length and sentiment score are treated as independent variables. The model's performance can be assessed using a variety of evaluators as discussed below.

6.4.1 Evaluation metrics

The models have been assessed in terms of accuracy, area under the curve, precision, recall, f1-score, kappa score and Mathew's correlation coefficient (Gupta & Rana, 2020).

1. **Accuracy**: It is the most widely used performance metric and is calculated as the number of correct predictions over all predictions (Sidhu, Kumar, & Rana, 2020), through Equation 6.5.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.5}$$

   where, TP stands for true positive, TN stands for true negative, FP stands for false positive and FN stands for false negative.

2. **Area Under the Curve**: The plot of sensitivity versus (1-specificity) is given by the Receiver Operating Characteristic curve. AUC converts the curve to a numeric value. The ranges of the curve and their corresponding interpretations are grouped as excellent for range varying from 1 to 0.90; good from 0.90 to 0.80; fair from 0.80 to 0.70; poor from 0.70 to 0.60 and fail from 0.60 to 0.50.

3. **Precision**: Precision is given through Equation 6.6 as:

$$Precision(P) = \frac{TP}{TP + FP} \tag{6.6}$$

4. **Sensitivity**: Sensitivity is the ratio of actually true classes that are identified correctly. Another name for sensitivity is true positive rate or recall. To reframe, it measures how often true predictions are correct. It is defined through Equation 6.7 as:

$$Recall(R) = \frac{TP}{TP + FN} \tag{6.7}$$

5. **F1 score**: It's an accuracy metric that considers the trade-off between precision and recall and is calculated through Equation 6.8 as:

$$f - measure(f) = 2 * \frac{P * R}{P + R} \tag{6.8}$$

6. **Kappa**: The Kappa score handles multi-class as well as imbalanced class

problems. It is defined through Equation 6.9 as:

$$Kappa = \frac{p_o - p_e}{1 - p_e} \tag{6.9}$$

where, p_o and p_e denote the observed and expected agreement, respectively. In general, it reflects how a classifier performs as compared to another classifier that simply guesses at random based on each class's frequency. Cohen's kappa is never greater than 1. When the value of kappa is zero, the classifier is useless.

7. **Matthews Correlation Coefficient (MCC)**: The Matthews correlation coefficient assesses the quality of a binary classification problem; it is a balanced measure for an unbalanced dataset as well. It outputs a value between minus one and plus one where, plus one indicates complete agreement between predicted and observed value, minus one indicates total disagreement, and zero value indicates random predicted values. It is defined through Equation 6.10 as:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6.10}$$

6.4.2 Performance analysis

The models have been assessed in terms of above evaluation metrics namely accuracy, area under the curve, recall, precision, f1-score, kappa and MCC.

Table 6.4: Performance of Machine Learning Models

Model	Accuracy	AUC	Recall	Precision	F1-Score	Kappa	MCC	TT(sec)
LR	0.99	0.99	0.99	0.99	0.99	0.99	0.99	11.5
DT	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.19
RF	0.99	0.99	0.99	0.99	0.99	0.99	0.99	2.47
ADA	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.23
GB	0.99	0.99	0.99	0.99	0.99	0.99	0.99	6.48
ET	0.9657	0.9995	0.9997	0.9612	0.98	0.8596	0.8683	9.14
KNN	0.9263	0.9471	0.9912	0.9263	0.9576	0.6762	0.7004	1.92
LDA	0.5788	0.517	0.6078	0.6427	0.6233	0.3348	0.3439	27.73

The resulting comparison is shown in Table 6.4.

As shown in Table 6.4 and Figure 6.2, most of the classification models perform decently when contrasted according to the evaluation parameters.

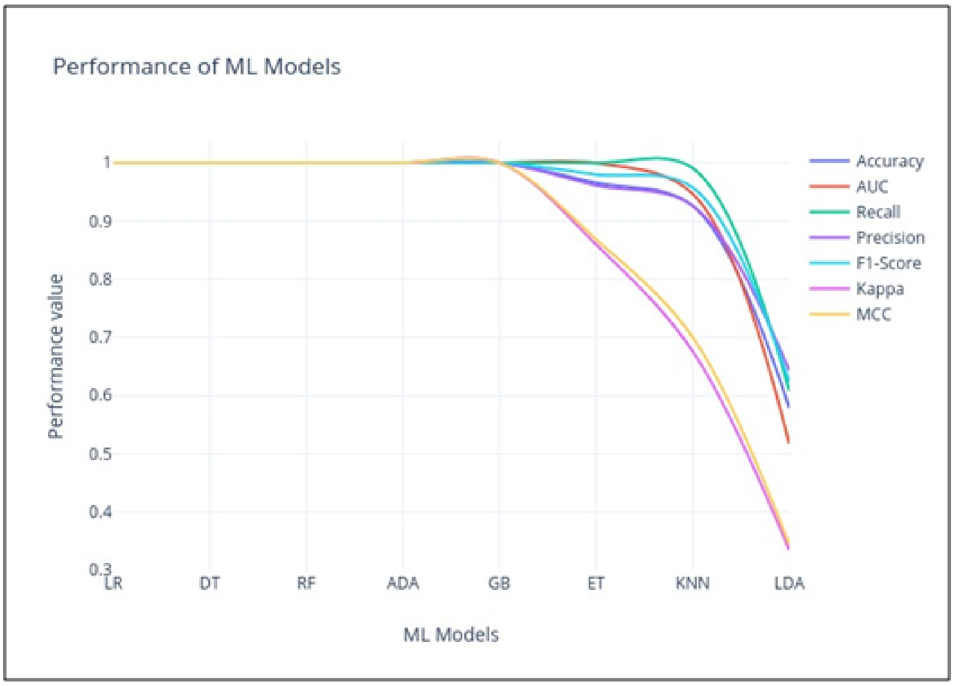

Figure 6.2: Performance of Machine Learning Models

In order to test the model's robustness, ten-fold cross-validation is employed. Due to lack of sufficient system RAM, the model is fed a random sample of 5000 rows, allowing for the above performance. Also, the methods' black-box state diminishes the results' interpretability. In comparison to others, LDA is unable to provide a reasonable prediction. The models have been trained again after performing feature selection and outlier removal to check the improvement in their performance. The near perfect performance of these models can be attributed to the size of data being fed to these models. Decision Tree model takes the least amount of time i.e. 0.19 seconds for training its model.

Upon performing feature selection, the accuracy of LDA model jumps to 0.8411, AUC increases to 0.732, recall, precision, f1-score, kappa and MCC turn out to be 0.892, 0.842, 0.866, 0.638 and 0.658 respectively. The threshold value used for feature selection is set to 0.8 and the classic method of permutation feature importance techniques is used. Even after performing feature selection, the performance of LR, DT, RF, ADA, and GB classifiers remains unaffected. The training time of all the models reduced. Training time of model- LR reduced to 6.32 from 11.5 (without feature selection), DT remained the same as 0.19, ADA classifier remained the same as 0.23, ET reduced to 9.05 from 9.14, KNN reduced to 1.90 from 1.92 and LDA reduced to 26.75 from 27. 73 seconds. Only two models RF and GB had their training time increased to 2.62 from 2.47 and 6.51 from 6.48 respectively.

Table 6.5: Performance of Machine Learning Models After Feature Selection

Model	Accuracy	AUC	Recall	Precision	F1-Score	Kappa	MCC	TT(sec)
LR	0.99	0.99	0.99	0.99	0.99	0.99	0.99	6.32
DT	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.19
RF	0.99	0.99	0.99	0.99	0.99	0.99	0.99	2.62
ADA	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.23
GB	0.99	0.99	0.99	0.99	0.99	0.99	0.99	6.51
ET	0.9634	0.99	0.99	0.9590	0.9790	0.8480	0.8580	9.05
KNN	0.9729	0.9910	0.9960	0.9730	0.9840	0.8920	0.8950	1.90
LDA	0.8411	0.7320	0.8920	0.8420	0.8660	0.6380	0.6580	26.75

Table 6.5 shows the performance of machine learning models after feature selection. Outliers from the training data have been reduced using Singular Value Decomposition and the outlier threshold has been set to 0.05, that is, five percent of the outliers have been removed from the training dataset. Again, the performance of LR, DT, RF, ADA, and GB classifiers remains unaffected. While the accuracy of ET and KNN classifiers increases, that of LDA decreases significantly. This implies that ET, KNN and LDA classifiers are affected due to removal of outliers whereas the rest of the classifiers are not affected with this processing step. Table 6.6 represents performance of classifiers after removal of outliers.

Table 6.6: Performance of Machine Learning Models After Outlier Removal

Model	Accuracy	AUC	Recall	Precision	F1-Score	Kappa	MCC	TT(sec)
LR	0.99	0.99	0.99	0.99	0.99	0.99	0.99	6.25
DT	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.18
RF	0.99	0.99	0.99	0.99	0.99	0.99	0.99	2.37
ADA	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.22
GB	0.99	0.99	0.99	0.99	0.99	0.99	0.99	6.42
ET	0.9759	0.9999	1.0000	0.9726	0.9861	0.8969	0.9019	6.43
KNN	0.9801	0.9938	0.9979	0.9793	0.9885	0.9168	0.9192	1.83
LDA	0.7523	0.7173	0.7673	0.8541	0.8060	0.4644	0.4877	23.48

As shown in Table 6.4, Table 6.5 and Table 6.6, LR, DT, RF, ADA and GB perform perfectly for the sample dataset provided to the models with and without feature selection and outlier removal process. The LDA model shows performance improvement after the feature selection process, but degradation after outlier removal and the accuracy of ET and KNN models improves after the removal of outliers. Also, considering TT(sec) i.e. training time to distinguish between the considered models, the Decision Tree model undergoes the training process most quickly.

6.5 CHAPTER SUMMARY

This chapter uses machine learning models, such as Logistic Regression, Decision Tree, Random Forest, Ada Boost, Gradient Boost, Extra Trees, K Nearest Neighbor, and Linear Discriminant Analysis, to identify helpful product reviews. Additional features have been derived through the process of feature engineering in addition to the characteristics that were present in the selected dataset. After performing preprocessing, feature engineering, model training, and model testing on the original and derived features, machine learning models are evaluated for accuracy, area under the curve, precision, recall, f1 score, Kappa score, and Mathew correlation coefficient metrics. These metrics are used to determine how well the models predict the useful reviews. LR, DT, RF, ADA and GB are performing perfectly for the sample dataset provided to the models with and without feature selection and outlier removal process. LDA model shows performance improvement after feature selection process, but degradation after outlier removal and accuracy of ET and KNN models improves after removal of outliers.

CHAPTER 7

PROPOSED METHOD TO UTILIZE REVIEW ANALYSIS IN RECOMMENDER SYSTEMS

CHAPTER VII

PROPOSED METHOD TO UTILIZE REVIEW ANALYSIS IN RECOMMENDER SYSTEM

This chapter describes the proposed method to utilize review analysis in recommender system. Interest of the user towards a product can be gauged in terms of the product's features through product reviews. This information is not available in recommender systems based only on product ratings. Hence, review based analysis to capture feature information and uninteresting item has been proposed.

7.1 INTRODUCTION

The digital era has enabled consumers to surf e-commerce web anytime and anywhere. Although accessing and surfing this domain has turned out to be a smooth sail, it's the overload of information that bogs down the actual purpose of consumer's search. The consumer is overloaded with humongous amount of alternatives for a product of interest, which makes it difficult for him/her to make a final decision call for purchasing the product. In order to downsize the information overload presented to the customer while trying to make a purchase, recommender system is used. Recommender system suggest items to a customer based on his or her interest. Success of such systems depends upon clicking an item for viewing its details to placing a purchase order for the item. E-commerce giant Amazon has been a pioneer in personalization and recommendation area (Kabir et al., 2019). Recommender systems take into account a customer's purchase history and his or her interests to provide him/her valuable recommendations. Of late, product reviews as a direct feedback mechanism, reflecting a customer's true experience with the product(s) have also been included in the process to generate effective recommendations

(Lin et al., 2017). These reviews are a first- hand reflection of consumer's encounter with his/her purchases. Emotional and sentiment analysis of these texts help to reflect people's mindset (Chehal, Gupta, & Gulati, 2020). Predicting users interest for items related to their interest with the help of reviews given for their past purchases can help identify uninteresting items. Previously, product reviews given by customers are used to generate recommendations. User opinions are collected from chat rooms or discussion platforms. Then, the review information is translated with the help of ontology and a new ranking mechanism is devised to rank the consumer's expertise in handling the products.

Even after having all the techniques and indicators and data such as rating data, behaviour pattern data, product data to the rescue of recommender system, such systems still face challenges like cold-start problem, data sparsity, data volatility, data volume, changing user preferences, synonymy, privacy, overspecialization (Adomavicius & Tuzhilin, 2005; Patel, Desai, & Panchal, 2017; Eirinaki, Gao, Varlamis, & Tserpes, 2018; Bunnell et al., 2019). It has been stated that ratings are a reflection of user's inclination towards a product. High rating value is given to item(s) of interest, whereas no rating is provided to uninteresting items. Accurate recommendations can be provided by injecting low rating values for unrated items. Unrated items are categorized into three:

1. No rating is provided when a user is unaware of the existence of the item
2. No rating is provided even when a user is aware of the existence of an item and had purchased the item
3. No rating is provided even when a user is aware of item's existence, but the user doesn't like the item, doesn't purchase the item and hence doesn't rate the item

The third one is the case of items of uninterest which implies unfavourable preference for such items. Ideally, a user should not be recommended these uninteresting items.

Usually, the ratings taken into consideration for an item are post-use preferences. But, the literature states the notion of user's pre-use preference (inclination towards an item before its purchase or use) for an item. Uninteresting items didn't receive any rating but are likely to receive insignificant or low user ratings. The previous methods imputed low rating for such unrated user-item pairs in the product rating matrix. The earlier CF approaches, evaluate all items with missing rating values as top-N recommendation candidates, and avoid uninteresting items as top N items recommendation candidates. Although the concept of uninteresting items is novel, however it finds its basis on star ratings. Star ratings reflect the experience of a user with respect to a scale, say 1 to 5. This type of feedback represents overall inclination of a user towards a product. A product's feature level preference of a user is not conveyed through star ratings. Thus, product reviews are preferred over star ratings as they are able to capture feature level preference of user. This study, identifies uninteresting items through review analysis in recommender system.

7.1.1 Topic modeling

Topic modeling is an unsupervised technique that aids in analyzing massive data sets. A document's underlying concepts are termed as topics. A topic is a group of words that frequently occur together, and a document is composed of topic(s). Clustering commonly occurring words is referred to as topic modelling (Abdelrazek, Eid, Gawish, Medhat, & Hassan, 2022). There exist three main methods for feature extraction, rule-based, sequential- based and topic model-based methods (Vamshi, Pandey, & Siva, 2018). A document-term matrix is what the topic modeling algorithm takes in as its input. With the use of a document-word matrix, documents are presented in the form of a bag-of-words model, in which the semantics and order of words are disregarded. The frequency with which individual words appear in a given document is the information that is contained in a document-word matrix. Modeling based on themes presupposes that each document has a variety of topics, each of which has a particular probability.

The Latent Dirichlet Allocation (LDA) method is a well-known topic modeling method based on the Bayesian theorem. In LDA, a document is generated through the combination of topics, and each topic is made up of individual words associated with specific probabilities. As a result, it is a probabilistic generative model, and its outputs include topics, the words that compose them, and the probabilities associated with those words. LDA is not limited to documents with unit topics as it can model long documents with several topics. However, LDA does not perform well when the input documents are of an insufficient length (Lin et al., 2017). With the use of LDA, a word that normally conveys one meaning can be associated with another topic as well (Abdelrazek et al., 2022). Yet, LDA is incapable of establishing links between different topics. For instance, the word ”mouse” can refer to both creatures and to a point-and-click interface on a computer. If a mouse is considered a device that is used with computers, then the discussion of hardware becomes more relevant.

7.2 PROPOSED METHODOLOGY

As part of the proposed methodology, uninteresting items have been identified with the help of feedback given in product reviews. The approach involves identifying product's features for which the customer has given negative feedback explicitly using sentiment analysis and also identify features that the customer does not prefer. This identification is based on the assumption that given the original product features say, X, reviewed features (either positive 'P' or negative 'N'), then uninteresting features or features not of interest for the user is given as features not reviewed FNR =(X-Y) features. The final recommendation list shouldn't include products with features for which the user has given negative feedback, 'N' and for features not reviewed 'FNR' by the user. The proposed method as shown in Figure 7.1 identifies uninteresting items with the help of feedback given by product reviews.

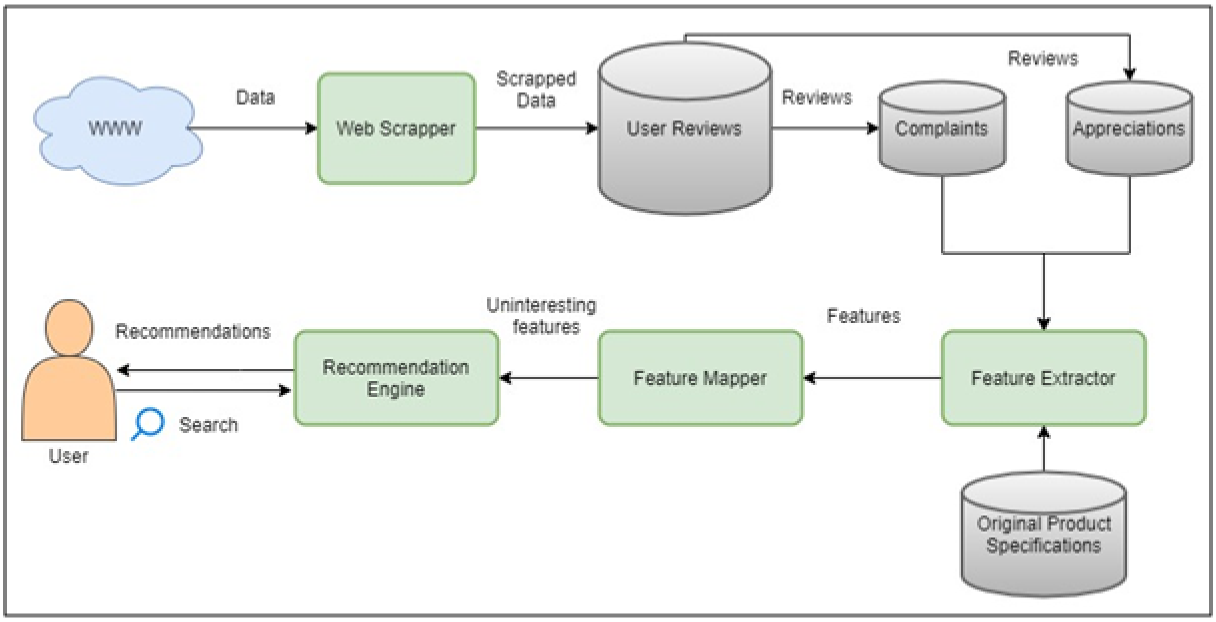

Figure 7.1: Proposed Method to Utilize Review Analysis in Recommender System

The steps of the proposed methodology are as follows:

1. **Data collection**: In the literature, publicly available reviews dataset provided by Amazon (*Amazon Customer Reviews Dataset*, 2022; *Amazon Product Data*, 2022) have been used. Scrapping technique has been used to download the product reviews from e-commerce website and build a unique dataset. A total of 269 reviews have been scrapped for Apple iPhone 11 Pro Max mobile. *BeautifulSoup* package in Python has been used to scrap the required data. The curated dataset consists of title of the review, date of review, content of the review and rating received by the product.

content	rating
Product is awesome. First time I am using iPhone. Plus points are battery life and user interface. It never fails to do tasks v	5.0 out
Best Phone yet	5.0 out
All is goodBut Apple is not giving the latest designs or technology in Apple they are giving simple design and all	4.0 out
Awesome phone	5.0 out
Excellent phone , Superb battery life and display quality outstanding..	5.0 out
Battery life is very good itƒ??s awesome I love 11pro max?????"	5.0 out
Trustable	5.0 out
It is good not better....	4.0 out
I will rate one start for this product. My experience of using this phone for one week is sub-par, multiple times in this one v	1.0 out
Very nice iphone	5.0 out
Your browser does not support HTML5 video. ?¥11 Pro Max 256 GB phone hanged last 12 hours continues	1.0 out
Pricey,but what you get is worth full. Buttery smooth in operation. Display is not over saturated.camera and battery life is	5.0 out
Only disliking feature is it's price	5.0 out

Figure 7.2: Snapshot of the Scrapped Dataset

Snapshot of the dataset scrapped is shown in Figure 7.2.

2. **Review classification into complaints and appreciations**: Post collection of data, the reviews have been segregated into complaints and appreciations using sentiment analysis (Vamshi et al., 2018). Sentiment analysis or opinion mining of user reviews helps to classify user opinion into positive, negative or neutral classes. In order to perform sentiment analysis, Python's *vaderSentiment* package has been utilized.

3. **Product features extraction**: In the next step, product's aspects are extracted from the complaints and appreciations using Latent Dirichlet Allocation topic modeling technique as shown in Figure 7.3. $'mobile','month','handy','love','big','phone','good','camera','phone', 'suit','storage','capacity','power','battery','good','phone','one'$

```
Python 3.7.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\Sandeep\AppData\Local\Programs\Python\Python37-32

['mobile', 'month', 'handy', 'love', 'big', 'phone', 'good', 'camera', 'phone',
'suit', 'storage', 'capacity', 'power', 'battery', 'good', 'phone', 'one']
>>>
```

Figure 7.3: Extraction of Features from Reviews

4. **Feature mapping**: In this step, features that have not been reviewed have been identified. Features not reviewed or FNR by the user are identified by map-

ping reviewed features with pre-defined product features. Pre-defined features information pertain to the original specification of the product.

5. **Recommendation generation**: The last step corresponds to generation of recommendations. Products from top N recommendation list with features as 'FNR' and negative features 'N' are removed so as to downsize the recommendation list and exclude uninteresting items from the suggestions.

The above mentioned steps have also been depicted as a flowchart in Figure 7.4:

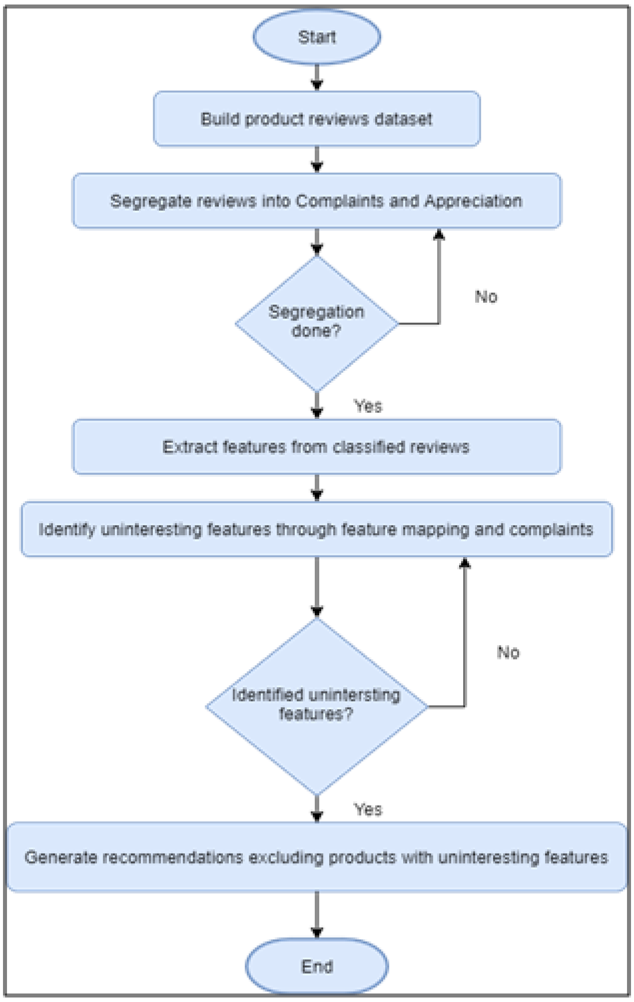

Figure 7.4: Flowchart Depicting Proposed Methodology

The first step of the proposed method is to build product reviews dataset. This is followed by categorizing of reviews into complaints or appreciations using sentiment analysis. In case a review has not been categorized into either of the two classes, the method is not proceeded further. Once the segregation is complete, feature extraction is performed using LDA topic modeling technique. Post feature extraction, mapping of original features from the product's specification and the features extracted from the previous step is done. Features which are part of complaints and are not reviewed by the user are considered to be uninteresting features. Once this identification is complete, recommendation list is generated. However, the products with uninteresting features identified as above are excluded from this generated recommendation list.

7.3 CHAPTER SUMMARY

Through this chapter an approach has been put forward that identifies products with uninteresting features for a user with the help of product reviews. This feature level information is not available in star ratings thereby highlighting the importance of inclusion of product reviews and their analysis in building recommender system. Products with uninteresting features should be refrained from including in the top N recommendation list of items. The future development of this approach shall consider inculcating deep learning and other topic modeling approaches to determine product features from user reviews.

CHAPTER 8

A STUDY ON THE IMPACT OF PANDEMIC ON E-COMMERCE

CHAPTER VIII

A STUDY ON IMPACT OF PANDEMIC ON E-COMMERCE

This chapter studies the impact of pandemic on e-commerce. The need to study this impact arises as during the imposed lockdowns functioning of e-commerce gets stalled. As a result, customer interaction gets minimised. As no customer feedback and existing studies on impact of pandemic are available, framing of policies and taking corrective measures for the benefit of all the stakeholders is delayed. Thus, such a study on the impact of pandemic on e-commerce domain has been contributed.

8.1 INTRODUCTION

A pandemic refers to a health crisis with a significant global spread (Yamin, 2020). Along with causing harm to human lives, pandemics damage the world economy. Recently, novel corona virus (2019-nCoV) spread in the entire world since its first human infection in December 2019 (Naserghandi, Allameh, & Saffarpour, 2020; Kamath, Kamath, & Salins, 2020). Many sectors have been hampered globally such as, aviation, automobile, education, oil industry, tourism, hospitality, real-estate, e-commerce etc. This leads to increased rates of unemployment and poverty also (Andrade Chittaranjan, 2020). India witnessed a rising number of infections since March 2020 (Venigalla, Chimalakonda, & Vagavolu, 2020; Aggrawal et al., 2021). Measures taken to combat the spread of this virus by the government of India include imposing of total lockdown for 21 days (25 March 2020 to 14 April 2020) so as to contain the spread of the virus (BBC, 2020); extending the lockdown till 3 May 2020 (also known as lockdown 2.0) (India Today, 2020); pushing the lockdown further with eased curbs (till 17 May 2020 also known as lockdown 3.0) (Jain, 2020); relaxed lockdown (till 31 May 2020 also known as lockdown 4.0); dividing the nation into green, orange and red zones; rapid testing of citizens in containment area (*Containment Plan for Large Outbreaks of COVID19*, 2020); mandatory wearing of masks (*Masks are mandatory for all now*, 2020) and social distancing among others.

Although the sentiment of Indians during the first lockdown was majorly positive with very less instances of disgust, anger and sadness (Prabhu, Kamath, & Pai, 2020; Barkur, Vibha, & Kamath, 2020). This research attempts to gauge their opinion after the initial lockdown was extended (from 15 April 2020 to 3 May 2020 aka lockdown 2.0) and compare them with those in lockdown 3.0 (from 4 May 2020 to 17 May 2020). Also, an attempt to discover latent topics related to e-commerce using Latent Dirichlet Allocation (LDA) topic modelling technique has been done. Recording the sentiment from time to time becomes essential for the government to take necessary actions. These actions/events impact the emotional well being of a person. The authorities after knowing citizen's emotional state can chalk out policies beneficial to them. Also, e-commerce stakeholders can adjust according to their state and regulate products demand and supply.

8.2 METHODOLOGY

This section describes the methodology followed to understand the users' mindset amid the pandemic. It consists of data collection and the steps followed. Data collection has been done daily through a freely available application programming interface API.

8.2.1 Data collection

Lockdown 2.0 period was marked with trending of events such as usage of Aarogya Setu app (through #AarogyaSetu); wear your mask challenge (#maskIndia); motivational song by famous Indian personalities (#muskurayegaIndia); mass gatherings for religious purpose (#TablighiJamaat), saluting the Coronavirus warriors of the country – police personnel, doctors, medical and health care staff (#IndiaSalutesCoronaWarriors); and facilitation of special trains for transition of labourers stuck in lockdown to their native places (#ShramikSpecialTrains) to name a few. While #IndiaFights Corona, #LiquorShops, #ShramikSpecialTrains, #Muslim Phobia_In_India, #HumModiKeSathHain, #Say_No_To_ Alcohol, #VijaySankalpAgainstCorona and #Aatma NirbharApnaBharat trended during lockdown 3.0. A total of 29,554 tweets of lockdown 2.0 from social media platform Twitter have been collected using 'twitteR' Application Programming Interface (API) by R. Tweets corresponding to trending hashtags were downloaded through R programming. A total of 47,672 tweets have been collected for the third lockdown and analysed using ‘syuzhet’ package in R for presence of emotions, namely, positive, negative, trust, fear, joy, anticipation, anger, sadness, surprise and disgust (Jockers, 2020).

8.2.2 Process

Each tweet can fall under different emotions and sentiments. Post data collection pre-processing has been done to clean the data as per the requirements.After pre-processing the tweets have been analysed using National Research Council Canada (NRC) emotion lexicon to study the feelings of Indians across eight different emotions and two sentiments during this period. This lexicon is a list of English words to which the emotions and sentiments are associated. The steps followed as part of pre-processing are as follows:

1. **Stop words removal**: Stop words such as ’a’, ’an’, ’the’ are frequently employed terms that do not significantly contribute to the semantics of any text. The elimination of stop words reduces its dimensionality. In texts, stop words are grammatical structures that provide no context. Prepositions, determiners, and coordinating conjunctions are included in the category of stop words.

2. **User mention removal**: User name of users posting a particular tweet is available in the tweets after symbol. This information is not relevant to the undertaken study and hence has been removed as part of the pre-processing step.

3. **Hyperlink removal**: User tweets often contain hyperlinks to external websites, such hyperlinks have been removed as they do not contribute to the sentiments of the user.

4. **Special characters removal**: All special characters and non-American Standard Code for Information Interchange (ASCII) characters except the hashtag have been removed from the refined tweet text. Also, all characters of length less than three alphabets have not been considered in tweet analysis (Symeonidis, Effrosynidis, & Arampatzis, 2018)

Further, tokenization and stemming using Porter Stemmer have been carried out to perform the opinion mining. Stemming is the basic text processing procedure that eliminates prefixes, infixes or suffixes from a word. The reduction of the conjugation forms of each word into its root is the goal of the stemming process. For instance, the English word ”complicate” can be modified with a morphological suffix to produce the phrases complication or complicating. It derives from the root word ”complicate.” Post pre-processing and stemming, NRC emotion lexicon has been used to generate word cloud, emotions and sentiments of the users (Mohammad & Turney, 2013). NRC emotion lexicon is a list of words in English language containing their manually annotated emotions and sentiments using crowdsourcing on Mechanical Turk. While the number of tagged sentiments are two, i.e. positive and negative, the number of tagged emotions are eight, namely, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. As per the latest update, using Google translate, the lexicon is available in more than hundred languages. The process flow to obtain these emotions and sentiment is shown in Figure 8.1.

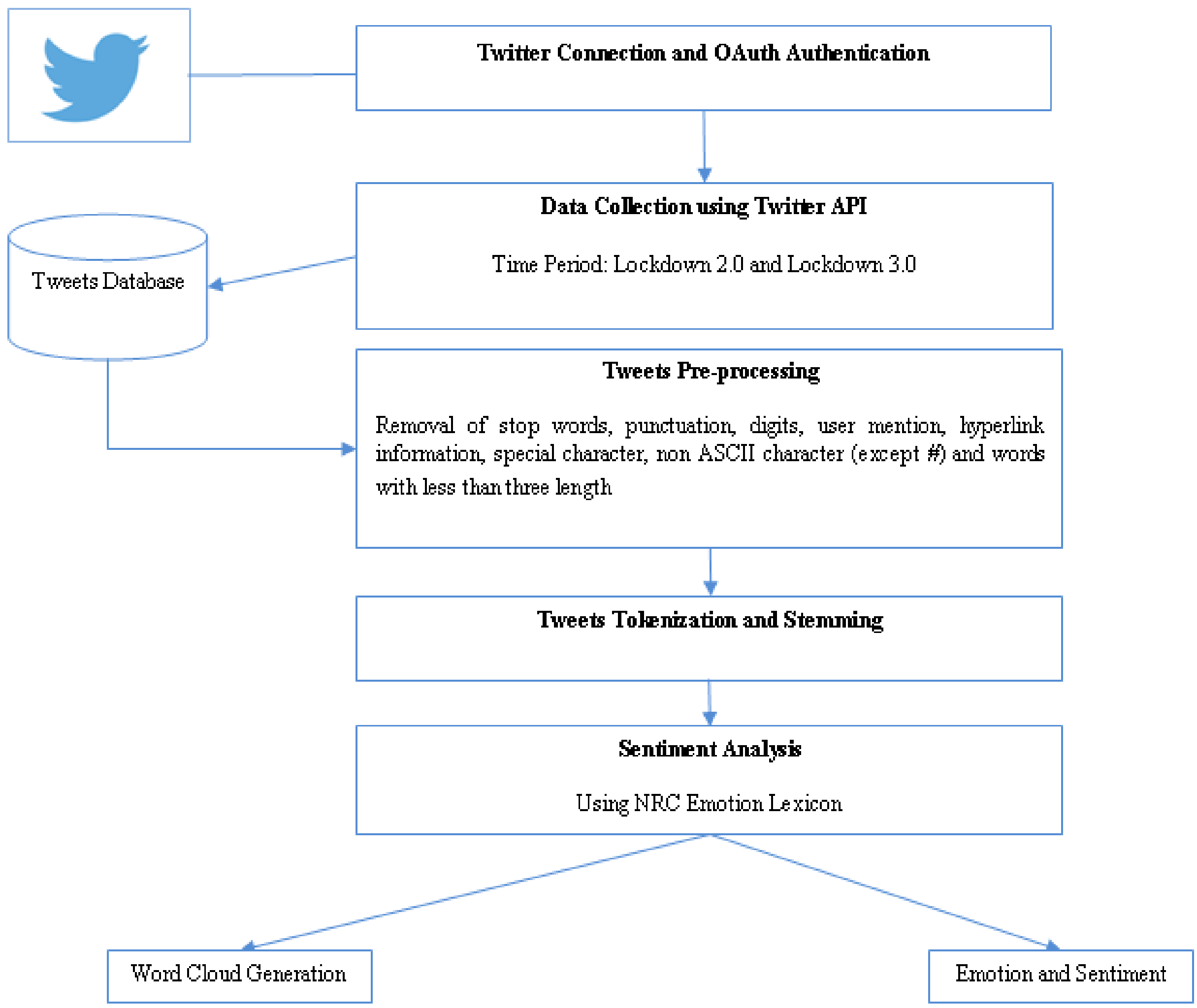

Figure 8.1: Flowchart of Sentiment Analysis

8.3 RESULT & DISCUSSION

This analysis has been performed using both Python and R language. As per the analysis performed, #AarogyaSetu has been the top positive hashtag in lockdown 2.0, whereas #IndiaFightsCorona has stood out as the top positive hashtag in lockdown 3.0.

The word cloud depicting the major emotions during lockdown 3.0 has been obtained as shown in Figure 8.2.

Figure 8.2: Word Cloud Analysis of Lockdown 3.0 Tweets in India

The word cloud shown above consists of all the eight emotions namely, anger, trust, surprise, sadness, joy, fear, disgust and anticipation. The size of the words in the word cloud depend on the frequency of occurrence. The more a word is frequent, the more is its font size. Each emotion's words have been highlighted in separate colors. The words related to anger emotion have been displayed in sky blue color and are listed as demand, lakh, capacity, video, package, leaders etc. The words related to trust emotion have been displayed in maroon color and are listed as modi, funds, business, yojana, education, app etc. The words related to surprise emotion have been displayed in blue color and are listed as speech, government, women, difficult, inspiring etc. The words related to sadness emotion have been displayed in green color and are listed as home, walk, states, govt, railways, trains etc. The words related to fear emotion have been displayed in purple color and are listed as mask, care, media, opened, time etc. The words related to disgust emotion have been displayed in pink color and are listed as muslim, civil, rss, mind, right etc. The words related to anticipation emotion have been displayed in red color and are listed as lockdown, leader, rahul, eid, proud, helping etc. Thus, the word cloud gives an overall idea of people's mindset during the pandemic. After word cloud generation, emotion and sentiment analysis has been performed. Through the analysis, the hashtags have been categorized into positive and negative classes.

Figure 8.3 depicts the top ten positive hashtags of lockdown 2.0.

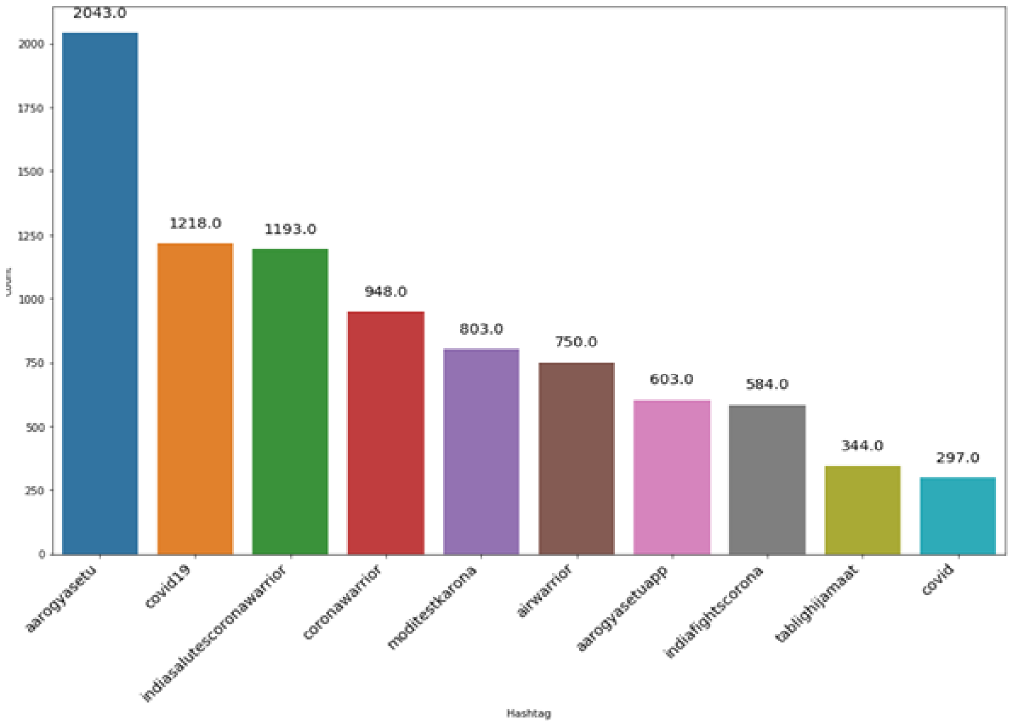

Figure 8.3: Top Ten Positive Hashtags during Lockdown 2.0 in India

Figure 8.4 depicts the top ten positive hashtags of lockdown 3.0.

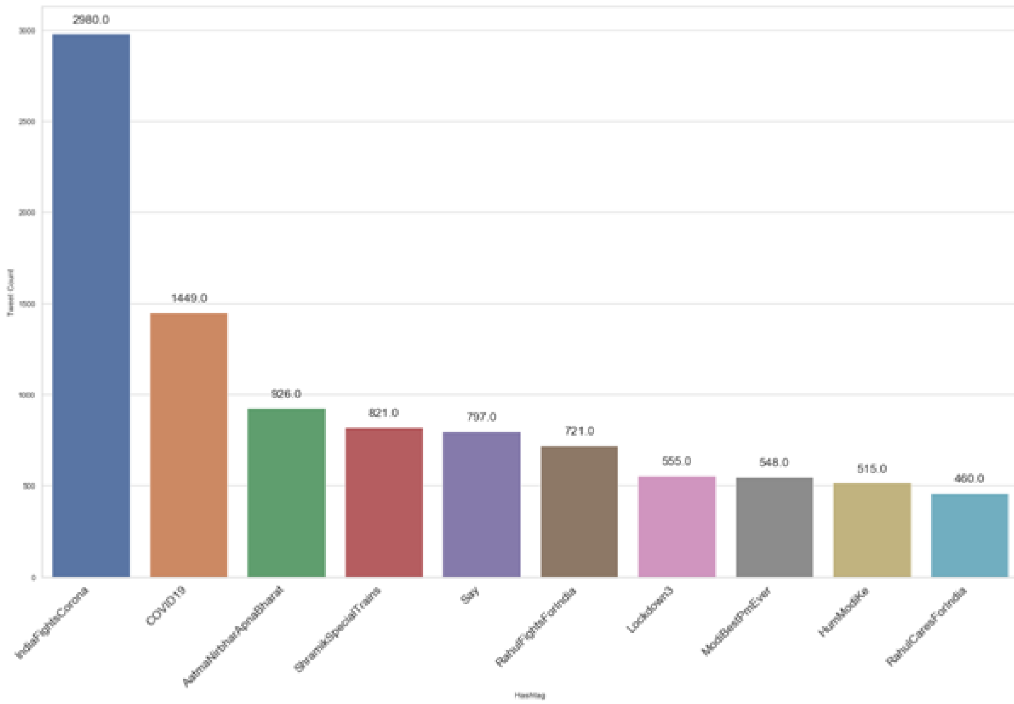

Figure 8.4: Top Ten Positive Hashtags during Lockdown 3.0 in India

The top negative hashtag has been #covid19 in lockdown 2.0 as shown in Figure 8.5.

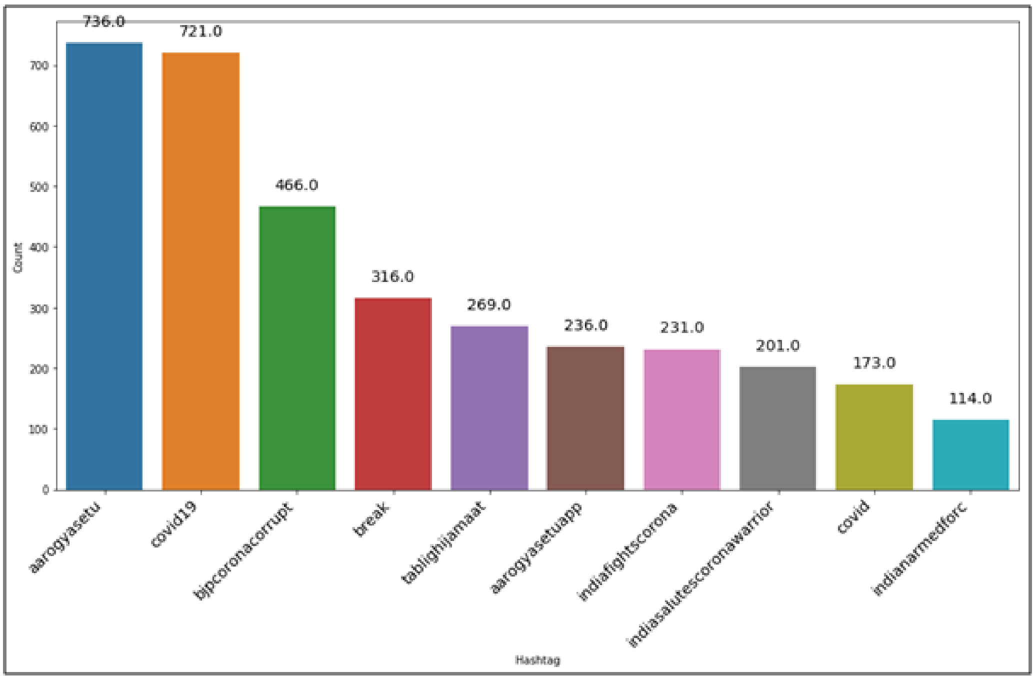

Figure 8.5: Top Ten Negative Hashtags during Lockdown 2.0 in India

The top negative hashtag has been #Say_No_To_Alcohol in lockdown 3.0 as shown in Figure 8.6. The number of positive tweets for #aarogyasetu, #aarogyasetuapp, #indiafightscorona and #indiasalutescoronawarrior have accounted more in number as compared to their negative tweets in lockdown 2.0.

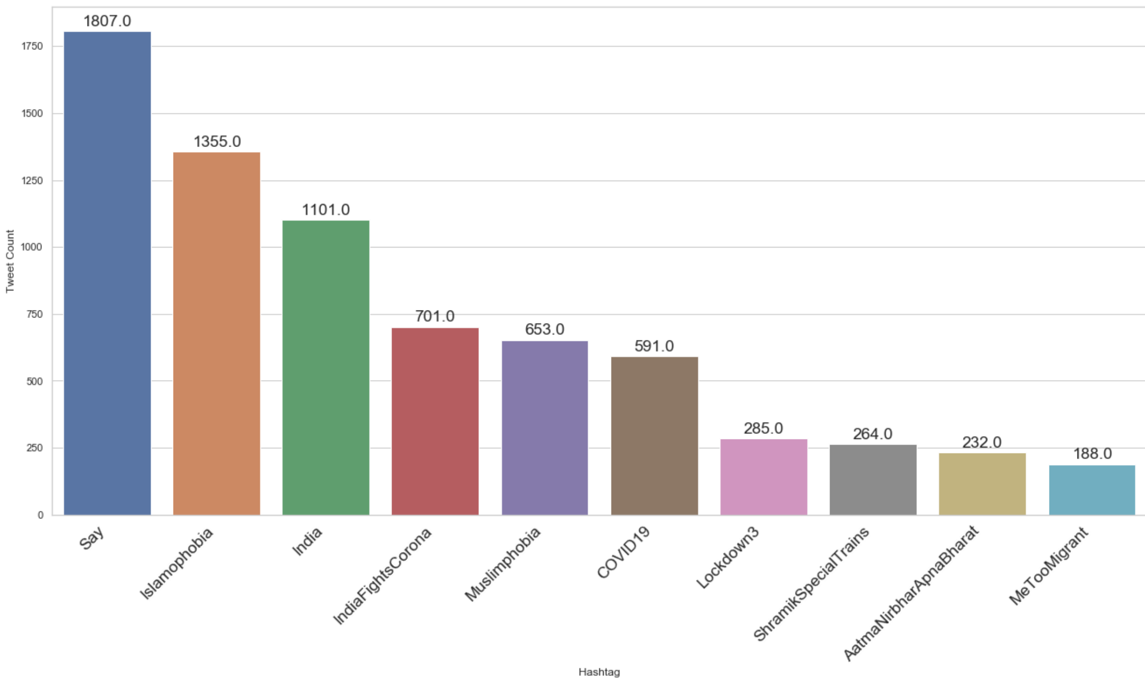

Figure 8.6: Top Ten Negative Hashtags during Lockdown 3.0 in India

The sentiment of Twitter users has been categorized into two sentiments, namely, positive and negative and eight emotions, namely, trust, fear, joy, anticipation, anger, sadness, surprise and disgust. Figure 8.7 represents the comparison of sentiment of twitter users in lockdown 2.0 and lockdown 3.0 for the above mentioned hashtags.

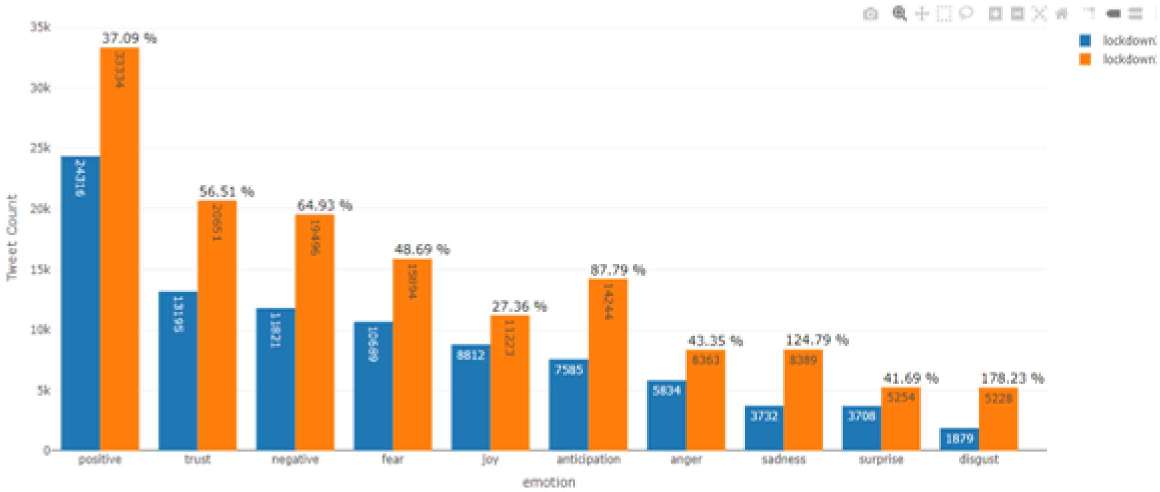

Figure 8.7: Emotional Analysis of Twitter Users-Lockdown 2.0 vs Lockdown 3.0

Disgust emotion witnessed the highest change in number of tweets (+178.23%) in lockdown 3.0 when compared with its number in lockdown 2.0. The second highest change in number of tweets (+124.79%) across the two lockdowns was witnessed for sadness emotion and the third highest change in number of tweets (+87.79%) was observed for anticipation emotion. This change in number of tweets in lockdown 3.0 from lockdown 2.0, when combined with the results suggest it can be due to #Say_No_To_Alcohol and #Islamophobia, the government, the opposition leader, respectively.

A tabular comparison of emotions and sentiments over lockdown 2.0 and lockdown 3.0 has been performed. While 'Decrease' depicts a dip in percentage of a particular emotion in lockdown 3.0 when compared with that in lock down 2.0, 'Increase' depicts a rise in this percentage. The following emotions – anger, fear, joy, surprise, trust and positive sentiment's percentage ((number of tweets associated with an emotion/total number of downloaded tweets in that lockdown) × 100) in lockdown 3.0 was less than that in lockdown 2.0. But emotions anticipation, disgust, sadness and negative sentiment's percentage in lockdown 3.0 was more than in lockdown 2.0. It should be noted that the total number of tweets at the bottom is not the numeric total of tweets across the emotions and sentiment but the total number of downloaded tweets for the lockdown period. Also, each tweet can be part of more than one emotion or sentiment due to the usage of NRC emotion lexicon.

This comparison is shown in Table 8.1 below. Social media cannot reflect the sentiment of the total population of any country, but it can surely be counted as a sample population to determine the vibe of a nation. The obtained analysis shows that not all went down well in the minds of citizens during the third lockdown. Maybe this is why the Indian government launched the financial stimulus package towards the end of lockdown 3.0 (Times, 2020).

Table 8.1: Emotional Analysis of Twitter Users-Lockdown 2.0 vs Lockdown 3.0

Emotion	**Lockdown 2.0**		**Lockdown 3.0**		**Trend**	**% change in no. of tweets**
Anger	5834	19.74%	8363	17.54%	Decrease	43.35
Anti-cipation	7585	25.70%	14244	29.88%	Increase	87.79
Disgust	1879	6.40%	5228	10.97%	Increase	178.23
Fear	10689	36.20%	15894	33.34%	Decrease	48.69
Joy	8812	29.80%	11223	23.54%	Decrease	27.36
Negative	11821	40.00%	19496	40.90%	Increase	64.93
Positive	24316	82.30%	33334	69.92%	Decrease	37.09
Sadness	3732	12.60%	8389	17.60%	Increase	124.79
Surprise	3708	12.50%	5254	11.02%	Decrease	41.69
Trust	13195	44.60%	20651	43.32%	Decrease	56.51
Total tweets down-loaded	29554		47672			

Understanding the shift in consumer behaviour during epidemics like Covid-19 also becomes vital so that the stakeholders involved can adjust to these changes and act swiftly in response to changing priorities. As per ShipBob's daily updates of e-commerce sales trends for merchants across verticals, baby products month over month (MoM) sales have surged online, while electronics week over week (WoW) sales have scaled up as on 16th April 2020 (*Daily Ecommerce Sales Trends & Resources— ShipBob 2020*, 2020).

Table 8.2 lists the MoM AND WoW sales across various e-commerce verticals by ShipBob. Amazon reported that its business in India is the most affected due to the Coronavirus lockdown. As per ShipBob's daily updates of e-commerce sales trends for merchants across verticals, baby products month over month (MoM) sales have surged online, while electronics week over week (WoW) sales have scaled up as on 16 April 2020. As shown, while baby products were the clear winner on 4 May 2020 due to the highest MoM sale percentage, it was apparel that stood first due to the highest MoM sale percentage as on 21 May 2020. Nutrition WoW sale percentage was the highest as on 21 May 2020 as compared with the rest of the e-commerce categories. This data reflects the consumer behaviour shifting from stocking up of baby products, sports and fitness products to apparel and nutrition, respectively.

Table 8.2: ShipBob's MoM and WoW E-Commerce Sales Trends

Vertical	**As on 4 May'2020**		**As on 21 May'2020**	
	MoM sale %	WoW sale	MoM sale %	WoW sale
Baby products	**693.9**	**-27.1**	**-72.9**	-10.01
Nutrition	-0.2	8.8	49.19	**46.96**
Food and Beverage	12.4	-6	15.9	**-28.62**
Beauty	64.6	62.7	48.73	-8.22
Apparel	20.4	62.5	**95.99**	6.93
Electronics	9.4	**71.7**	24.94	2.04
Toys & Games	66.5	21.9	44.57	-6.58
Sports &Fitness	112.2	-7.6	14.99	1.29
Jewellery	**-39.6**	0.9	13.57	2.33
Household goods	-2.4	2.8	72.09	11.05

8.3.1 Impact of COVID-19 on e-commerce and consumer shopping behaviour

Understanding the shift in consumer behaviour during epidemics like Coronavirus also becomes vital so that the stakeholders involved can adjust to these changes and act swiftly in response to changing priorities. During Corona times, consumers are worried about the delivery timelines being met by the e-commerce companies (*Daily*

Ecommerce Sales Trends & Resources— ShipBob, 2020), hygiene standards being followed as the last step in delivery reaches the customer's door and rise in prices of otherwise discounted products sold online once things return to normalcy among other concerns. Around 1,555 and 1,455 tweets were extracted using Twitter API to understand the latent information about e-commerce during lockdown 2.0 and lockdown 3.0, respectively in India. LDA topic modelling technique was implemented on the extracted tweets and the number of topics was set to three. As shown in Table 8.3, in lockdown 2.0, Topic 1 talks about Amazon and flipkart selling items online during lockdown. Topic 2 talks about delivery of processed, packaged and junk food to people during COVID-19. Topic 3 talks about requesting the government to allow Amazon India to deliver goods and products.

Table 8.3: Topics Generated from E-Commerce Lockdown 2.0 Tweets using Latent Dirichlet Allocation

Topic 1		**Topic 2**		**Topic 3**	
Word	Prob.	Word	Prob.	Word	Prob.
India	0.017	fake	0.041	AmazonIndia	0.040
AmazonIndia	0.017	deliver	0.041	goods	0.038
amazon	0.015	Paytm	0.041	delivery	0.029
flipkart	0.015	Walmart	0.040	Allow	0.028
items	0.014	processed	0.039	Government	0.027
online	0.01	packaged	0.039	Commerce	0.027
With	0.009	junk food	0.039	Products	0.025
sell	0.009	con- tributes	0.039	Including	0.025
like	0.008	COVID	0.021	Offers	0.025
lock down	0.008	people	0.017	Requests	0.025

In lockdown 3.0, Topic 1 talks about delivering of the product- Motorola Edge, non-governmental organization (NGO), AmazonIN and Flipkart. Topic 2 talks about delivery of essential products in lock - down by flipkart. Topic 3 talks about Quiz time mornings with Amazon, a quiz by Amazon for its customers, as shown in Table

8.4:

Table 8.4: Topics Generated from E-Commerce Lockdown 3.0 tweets using Latent Dirichlet Allocation

Topic 1		**Topic 2**		**Topic 3**	
Word	Prob.	Word	Prob.	Word	Prob.
India	0.032	flipkartsupport	0.025	amazon in-dia	0.023
FlipkartStories	0.021	order	0.02	eligible	0.014
product	0.017	delivery	0.013	QuizTime Morn-ingsWith Amazon	0.013
Moto-rola	0.017	amazon	0.012	Link	0.012
deliver	0.016	amazonIN	0.011	quiz	0.012
NGOs	0.015	time	0.01	India	0.011
order	0.014	product	0.01	Offer	0.011
price	0.014	essential	0.009	available	0.009
amazonIN	0.011	lockdown	0.009	AmazonSpin andWin	0.009
Edge	0.011	commerce	0.009	sale	0.009

8.4 CONCLUSION

Through this study, an attempt has been made to understand the mind-set of Indian people during the pandemic using Python and R statistical software. The tweets collected for this study are in English language which might serve as a limitation for the study. Also, the tweet collection has been done after every week as the free twitter API provided access to tweets from the last 7 days only. As the conversion rates have declined and consumer confidence has gone for a fall, it is time for the e-commerce giants to strategise. While some measures have been taken like sale of essential items, sale of nonessential items, there is a lot more to be done to boost this sector's growth. As the lockdown curbs will be eased in lockdown 4.0, sales are predicted to boost up with delivery restrictions still in place for areas categorized in red zones. The ultimate objective of the ecommerce industry should be uninterrupted and timely availability of indispensable products to prevent panic among customers. This study highlights the changing mind-set of people through the lockdowns. A significant number of tweets were witnessed for disgust, sadness and anticipation emotions indicating that the authorities need to buck up. This analysis can help the health specialists to understand people's mind-set, the authorities to take further corresponding measures in washing out the virus and the e-commerce stakeholders to adapt to the changing attitudes by adjusting demand and supply plans accordingly.

8.5 CHAPTER SUMMARY

A study on the impact of pandemic on e-commerce domain has been contributed. An attempt has been made to understand the mind-set of Indian people using Python and R statistical software, during the recent lockdown 2.0 (15 April 2020 to 3 May 2020) and lockdown 3.0 (4 May 2020 to 17 May 2020) through their tweets on the social media platform Twitter. The need to study this impact arises as during the imposed lockdowns functioning of e-commerce gets stalled. As a result, customer interaction gets minimised. Also, opinion on e-commerce during this pandemic has been analysed. Although the country had a positive approach in lockdown 2.0 with only a few instances of sadness, disgust and others, the majority of the people had a negative approach in lockdown 3.0. This analysis can help the health specialists to understand people's mind-set, the authorities to take further corresponding measures in washing out the virus and the e-commerce stakeholders to adapt to the changing attitudes by adjusting demand and supply plans accordingly.

CHAPTER 9

CONCLUSION AND FUTURE WORK

CHAPTER IX

CONCLUSION AND FUTURE WORK

This chapter summarizes the dissertation work carried out and provides directions for conducting future research. This research has contributed solutions corresponding to research objectives related to designing of e-commerce recommender system using data analysis.

9.1 CONCLUSION

Recommender system is a significant tool that assists users in their purchase decision by offering item suggestions relevant to their preferences. A notable level of research has been carried out to improve this system since mid 1990s. However, the system grapples with several limitations such as the famous cold-start and sparsity problem. The main step in collaborative filtering based recommender system is similar user or item identification. Previous research studies focus on identification of similar users through product ratings. Attribute-level information is not available in star ratings. As a result, the user/product profile built through the help of user reviews can be used to augment the available product ratings. Also, interest of the user towards a product/service can be gauged in terms of the product's/service's features. This feature-based information is not available in recommender systems based only on product ratings. Hence, there is a need to build review based recommender system to capture attribute level information.

Further, building customer and product profiles through customer reviews and identifying similar users through product ratings creates a gap that requires modification of existing processes.

Also, issues such as data sparsity affects the performance of recommender systems. Side information such as customer reviews have been used in the past to tackle the sparsity problem. However, due to subjectivity of reviews, a new sparsity problem is generated. There is a need to alleviate this novel sparsity problem to improve the recommender system performance.

In recommender system, existing product reviews are leveraged by other users to arrive at a purchase decision. However, browsing of large number of reviews is required to avail this benefit. Although, reviews are associated with users' vote to indicate their helpfulness, newly posted reviews do not receive these votes. Additionally not all reviews are tagged with such kind of votes by the users due to factors such as humongous volume of electronic word of mouth, voluntary helpfulness voting mechanism, level of visibility and their recentness. This necessitates the need to predict the review usefulness.

Lastly, during imposed lockdowns in the middle of pandemic, all domains functioning including e-commerce gets stalled. As a result, customer interaction gets minimised. As no customer feedback and existing study on impact of pandemic is available, framing of policies and taking corrective measures for the benefit of all the stakeholders is delayed. This gives rise to a need to study the impact of pandemic on e-commerce domain.

The proposed work achieves the objectives associated to the above issues as follows:

1. **Similar user identification through customer reviews**: A new approach to identify similar users by finding their sentiment for an item hidden in textual reviews is proposed. The proposed system first calculates the user sentiment score for each item and then finds the user similarity with other users who have reviewed the same set of items. At the end, using both the above scores, the sentiment score for each item by each user is then predicted. This approach can be used to utilise the hidden sentiment stored in the form of text in user reviews as an input to collaborative filtering technique. As an improvement, in the future, this work can be tested on huge datasets to verify if the method is scalable.

2. **Alleviation of sparsity problem in recommender system**: User reviews which form one of the important sources of side information have been leveraged to improve the performance of recommender system. Aspect based sentiment analysis (ABSA) of product reviews helps in identifying the contributing aspect(s) and their corresponding polarity, thereby providing a more detailed analysis of customer's inclination towards feature(s) of a product. An annotated dataset has been provided for performing aspect level sentiment analysis of mobile phone (Apple iPhone 11) reviews given by customers in English language on popular e-commerce website Amazon.The data has been

scrapped from Amazon India website using Python's BeautifulSoup package and annotated manually with predefined aspect categories and aspect sentiments. Naïve Bayes, Support Vector Machine, Logistic Regression, Random Forest, K- nearest Neighbour, Multi Layer Perceptron and a deep learning model -Keras Sequential Model API have been used to support the accuracy of the dataset. Deep Learning Model- Keras Sequential Model for classifying review text into 15 predefined aspect categories produces the most accurate result with an accuracy of 67.45%. While K- nearest neighbour fares the lowest in this task with only 49.92% accuracy. Multi Layer Perceptron's accuracy is the highest for classifying review text into 3 predefined aspect sentiments with an accuracy of 80.41%. While that of the Sequential model is the lowest with 76.30%. Accuracy is obtained for mostly all the ML models in the range of 49-67% in case of aspect category classification and 76 to 80% for the collected datasets in case of aspect sentiment classification. As satisfactory accuracy has been obtained for the collected dataset, it can be used as a benchmark dataset for ABSA of mobile phone reviews in English language.

In this contribution, understanding the intent conveyed by emoticons has not been considered. Also, the abbreviated words have not been handled in any special manner. The dataset collected for the single entity – Apple-iPhone11 mobile Phone has less than 1000 reviews which generates a small corpus of labelled dataset but with significant results. The results generated above are on the actual imbalanced data collected which can be further improved by balancing the dataset. Keras Sequential model, a deep learning model is the most accurate when the number of predefined aspect categories was fifteen and is the least accurate when the number of predefined aspect sentiments is three indicating the need for more data for the machine learning training process. Traditional ML model Multi Layer Perceptron performs the best when only three predefined aspect sentiments are to be classified.

A new kind of sparsity problem originating due to subjectivity of reviews has been explained and alleviated with the help of matrix factorization and autoencoder technique. The experiments conducted reveals that performance of the recommender system after removal of sparsity is the best and improves with high margins. Root Mean Square Error, Mean Absolute Error, Mean Squared Error of collaborative filtering recommendation algorithms with non-sparse input shows significant improvements thereby justifying the proposed approach in this study. The study is limited by processing power therefore, in future, this method can be implemented in a parallel or distributed fashion for faster and time-saving results. Also, the entire recommendation system's accuracy can be considered using some aspect based similarity measure instead of the conventional similarity measures.

3. **Determination of useful reviews**: Using the Amazon product review dataset of cell phones, machine learning models are built on eight features namely, overall, reviewtext, summary, and vote, as well as derived features such as word count, character count, average word count, and sentiment score and compared on seven performance measures, accuracy, area under the curve, precision, recall, f1-score, Kappa score and Mathews Correlation Coefficient. As per results, all the classification models perform well, except Linear Discriminant Analysis. The classification performance of Logistic Regression, Decision Tree, Random Forest, Ada Boost, and Gradient Boost is unaffected by feature selection or outlier removal. The performance of Linear Discriminant Analysis improves after feature selection but decreases after outlier removal, whereas Extra Tress and K Nearest Neighbour classifiers improves in both cases. The results of this research can assist e-commerce platforms in gaining a better understanding of the usefulness of online reviews. They can automatically analyse the usefulness of product reviews by utilizing prediction models as stated above. This study is limited due to lack of sufficient system RAM, the models are fed a random sample of 5000 rows. Also, the methods' black-box state diminishes the results' interpretability. The study can be strengthened by improving the prediction models by removing fake reviews, incorporating emoticons for online review helpfulness prediction, employing unsupervised learning techniques instead of supervised learning, and developing deep learning model.

4. **Review analysis incorporation in recommender system**: An approach to leverage the product review functionality has been proposed. Extracting the product features using Latent Dirichlet Allocation topic modelling technique from user-provided feedback and not recommending products with uninteresting product features to improve the recommendation list is the main idea behind this approach. Product with uninteresting features is identified using sentiment analysis and feature mapping. Such products are refrained from including in the top N recommendation list of items. The future development of this approach considers inculcating deep learning approach to determine product features from user reviews. Also, such a system shall be evaluated with the help of users.

5. **Analysis of user inclination towards e-commerce**: The contribution highlights the changing mindset of people through the lockdowns in pandemic. A

significant number of tweets are witnessed for disgust, sadness and anticipation emotions indicating that the authorities need to come up with improved strategies. Also, the tweets which are collected for this study are in English language which might serve as a limitation for the study. Also, the tweet collection has to be done after every week as the free twitter API provides access to tweets from the last seven days only. As the conversion rates have declined and consumer confidence has gone for a fall, it's time for the e-commerce giants to strategise. While some measures have been taken like sale of essential items, sale of non-essential items, there's a lot more to be done to boost this sector's growth. The ultimate objective of the ecommerce industry should be uninterrupted and timely availability of indispensable products to prevent panic among customers.

9.2 SCOPE FOR FUTURE WORK

In this work solutions to various problems in the recommender system have been proposed. However, the solutions have some limitations as well which have been mentioned corresponding to each solution above. As an extension to the research work carried out, the following can be incorporated to design a more transparent and scalable recommender system with improved performance:

1. **Fake reviews identification**: Fake reviews are the product reviews which are posted by the users without purchasing the product or actually having an experience with the product. Paid promotion or demotion of a certain product is often the reason behind such fake reviews. Inclusion of such reviews in data analysis shall lead to a certain amount of bias in the recommendation generation process. It becomes imperative to identify such fake reviews and exclude them after careful analysis for a bias free recommendation generation process.

2. **Recommendation justification**: A recommender system that provides justification or explanation of recommendations given to a user is a more transparent recommender system than a traditional recommender system. Justifying recommendations helps the user gain a better understanding of the recommended item thereby making the system more transparent.

3. **Scalable recommender system**: The proposed solutions to the research problems shall be tested for scalability on a framework that supports management of big data.

REFERENCES

REFERENCES

Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., & Hassan, A. (2022). Topic modeling algorithms and applications: A survey. *Information Systems*, 102131.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. doi: 10.1109/TKDE.2005.99

Aggrawal, P., Jolly, B. L. K., Gulati, A., Sethi, A., Kumaraguru, P., & Sethi, T. (2021). Psychometric Analysis and Coupling of Emotions between State Bulletins and Twitter in India during COVID-19 Infodemic. *Frontiers in Communication*, *6*(July 2017). doi: 10.3389/fcomm.2021.695913

Ahmadian, S., Afsharchi, M., & Meghdadi, M. (2019). A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems. *Multimedia Tools and Applications*, *78*(13), 17763–17798. doi: 10.1007/s11042-018-7079-x

Akbarabadi, M., & Hosseini, M. (2020). Predicting the helpfulness of online customer reviews: The role of title features. *International Journal of Market Research*, *62*(3), 272–287. doi: 10.1177/1470785318819979

Akhtar, M. S., Ekbal, A., & Bhattacharyya, P. (2018). Aspect based sentiment analysis: category detection and sentiment classification for hindi. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9624 LNCS*, 246–257. doi: 10.1007/978-3-319-75487-1_19

Alabdulrahman, R., & Viktor, H. (2021). Catering for unique tastes: Targeting greysheep users recommender systems through one-class machine learning. *Expert Systems with Applications*, *166*(September 2020), 114061. Retrieved from `https://doi.org/10.1016/j.eswa.2020.114061` doi: 10.1016/j.eswa.2020.114061

Aljuhani, S. A., & Alghamdi, N. S. (2019). A comparison of sentiment analysis methods on Amazon reviews of Mobile Phones. *International Journal of Advanced Computer Science and Applications*, *10*(6), 608–617. doi: 10.14569/ijacsa.2019.0100678

Alqadah, F., Reddy, C. K., Hu, J., & Alqadah, H. F. (2015). Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. *Knowledge and Information Systems*, *44*(2), 475–491. doi: 10.1007/s10115-014-0771-x

Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018). Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. *Journal of Computational Science*, *27*, 386–393. Retrieved from `http://dx.doi.org/10.1016/j.jocs.2017.11.006` doi: 10.1016/j.jocs.2017.11.006

Al-Smadi, M., Qawasmeh, O., Talafha, B., & Quwaider, M. (2015). Human Annotated Arabic Dataset of Book Reviews for Aspect Based Sentiment Analysis. *Proceedings - 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015*, 726–730. doi: 10.1109/FiCloud.2015.62

Alsubari, S. N., Deshmukh, S. N., Al-Adhaileh, M. H., Alsaade, F. W., & Aldhyani, T. H. (2021). Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets. *Applied Bionics and Biomechanics*, *2021*. doi: 10.1155/2021/5522574

Amazon Customer Reviews Dataset. (2022). Retrieved from `https://s3.amazonaws.com/amazon-reviews-pds/readme.html,year=2022`

Amazon Product Data. (2022). Retrieved from `http://jmcauley.ucsd.edu/data/amazon/,year=2022`

Andrade Chittaranjan. (2020). Understanding delayed effects on health of COVID-19 & lockdown. *Indian Journal of Psychiatry*, 247–249. Retrieved from `https://www.deccanherald.com/opinion/understanding-delayed-effects-on-health-of-covid-19-lockdown-825968.html`

Ardimansyah, M. I., Huda, A. F., & Baizal, Z. K. (2018). Preprocessing matrix factorization for solving data sparsity on memory-based collaborative filtering. *Proceeding - 2017 3rd International Conference on Science in Information Technology: Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech 2017*, *2018-Janua*, 521–525. doi: 10.1109/ICSITech.2017.8257168

Arif, M., Qamar, U., Khan, F. H., & Bashir, S. (2019). A Survey of Customer Review Helpfulness Prediction Techniques. In *Advances in intelligent systems and computing* (Vol. 868, pp. 215–226). Springer International Publishing. Retrieved from `http://dx.doi.org/10.1007/978-3-030-01054-6{_}15http://`

link.springer.com/10.1007/978-3-030-01054-6{_}15 doi: 10.1007/978-3-030-01054-6_15

Ayub, M., Ghazanfar, M. A., Mehmood, Z., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Unifying user similarity and social trust to generate powerful recommendations for smart cities using collaborating filtering-based recommender systems. *Soft Computing*, *24*(15), 11071–11094. Retrieved from https://doi.org/10.1007/s00500-019-04588-x doi: 10.1007/s00500-019-04588-x

Barkur, G., Vibha, & Kamath, G. B. (2020). Sentiment analysis of nationwide lockdown due to COVID 19 outbreak: Evidence from India. *Asian Journal of Psychiatry*, *51*(April), 102089. Retrieved from https://doi.org/10.1016/j.ajp.2020.102089 doi: 10.1016/j.ajp.2020.102089

Barrière, V., & Kembellec, G. (2018). Short review of sentiment-based recommender systems. *ACM International Conference Proceeding Series*, 6–9. doi: 10.1145/3240117.3240120

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, *52*, 1–37.

BBC. (2020). *Coronavirus: India enters 'total lockdown' after spike in cases* (No. March 2020). Retrieved from https://www.bbc.com/news/world-asia-india-52024239

Bilal, M., Marjani, M., Hashem, I. A. T., Gani, A., Liaqat, M., & Ko, K. (2019). *Profiling and predicting the cumulative helpfulness (Quality) of crowd-sourced reviews* (Vol. 10) (No. 10). doi: 10.3390/info10100295

Bilal, M., Marjani, M., Hashem, I. A. T., Malik, N., Lali, M. I. U., & Gani, A. (2021, jan). Profiling reviewers' social network strength and predicting the "Helpfulness" of online customer reviews. *Electronic Commerce Research and Applications*, *45*, 101026. doi: 10.1016/j.elerap.2020.101026

Bunnell, L., Osei-Bryson, K. M., & Yoon, V. Y. (2019). RecSys Issues Ontology: A Knowledge Classification of Issues for Recommender Systems Researchers. *Information Systems Frontiers*. doi: 10.1007/s10796-019-09935-9

Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, *5*(1). doi: 10.1145/1921591.1921593

Chehal, D., Gupta, P., & Gulati, P. (2020, jul). COVID-19 pandemic lockdown: An emotional health perspective of Indians on Twitter. *International Journal of Social Psychiatry*, 002076402094074. Retrieved from http://journals.sagepub.com/doi/10.1177/0020764020940741 doi: 10.1177/

0020764020940741

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, *25*(2), 99–154. doi: 10.1007/s11257-015-9155-5

Cheng, Z., Chang, X., Zhu, L., Kanjirathinkal, R. C., & Kankanhalli, M. (2019). MMalfM: Explainable recommendation by leveraging reviews and images. *ACM Transactions on Information Systems*, *37*(2), 1–28. doi: 10.1145/3291060

Chollet, F. (2015). *GitHub - keras-team/keras: Deep Learning for humans.* Retrieved from `https://github.com/keras-team/kerashttps://github.com/keras-team/keras{%}0Ahttps://github.com/keras-team/keras/{%}0Ahttps://github.com/keras-team/keras`

Chollet, F. (2020). *The Sequential model.* Retrieved 2020-09-10, from `https://keras.io/guides/sequential{_}model/`

Containment Plan for Large Outbreaks of COVID19. (2020). Retrieved from `https://www.mohfw.gov.in/pdf/3ContainmentPlanforLargeOutbreaksofCOVID19Final.pdf`

Daily Ecommerce Sales Trends & Resources— ShipBob. (2020). Retrieved from `https://trends.shipbob.com/{#}lp-pom-block-1404`

Daily Ecommerce Sales Trends & Resources— ShipBob 2020. (2020). Retrieved from `https://trends.shipbob.com/{#}lp-pom-block-1404`

da Silva, J. F. G., de Moura Junior, N. N., & Caloba, L. P. (2018). Effects of Data Sparsity on Recommender Systems based on Collaborative Filtering. *Proceedings of the International Joint Conference on Neural Networks*, *2018-July*. doi: 10.1109/IJCNN.2018.8489095

Da'u, A., & Salim, N. (2020). *Recommendation system based on deep learning methods: a systematic review and new directions* (Vol. 53) (No. 4). Springer Netherlands. Retrieved from `https://doi.org/10.1007/s10462-019-09744-1` doi: 10.1007/s10462-019-09744-1

Da'u, A., Salim, N., Rabiu, I., & Osman, A. (2020). Weighted aspect-based opinion mining using deep learning for recommender system. *Expert Systems with Applications*, *140*. doi: 10.1016/j.eswa.2019.112871

del Carmen Rodríguez-Hernández, M., & Ilarri, S. (2021). AI-based mobile context-aware recommender systems from an information management perspective: Progress and directions. *Knowledge-Based Systems*, *215*. doi: 10.1016/j.knosys.2021.106740

Dey, D., & Kumar, P. (2019, dec). A novel approach to identify the determinants of online review helpfulness and predict the helpfulness score across product categories. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11932 LNCS, pp. 365–388). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-37188-3{_}21http://link.springer.com/10.1007/978-3-030-37188-3{_}21 doi: 10.1007/978-3-030-37188-3_21

Dragoni, M., Federici, M., & Rexha, A. (2019). An unsupervised aspect extraction strategy for monitoring real-time reviews stream. *Information Processing and Management*, *56*(3), 1103–1118. Retrieved from https://doi.org/10.1016/j.ipm.2018.04.010 doi: 10.1016/j.ipm.2018.04.010

Du, J., Rong, J., Wang, H., & Zhang, Y. (2021). Neighbor-aware review helpfulness prediction. *Decision Support Systems*(April), 113581. Retrieved from https://doi.org/10.1016/j.dss.2021.113581 doi: 10.1016/j.dss.2021.113581

Du, J., Zheng, L., He, J., Rong, J., Wang, H., & Zhang, Y. (2020). An Interactive Network for End-to-End Review Helpfulness Modeling. *Data Science and Engineering*, *5*(3), 261–279. Retrieved from https://doi.org/10.1007/s41019-020-00133-1 doi: 10.1007/s41019-020-00133-1

Duan, R., Jiang, C., & Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decision Support Systems*, *156*, 113748.

Eirinaki, M., Gao, J., Varlamis, I., & Tserpes, K. (2018). Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions. *Future Generation Computer Systems*, *78*, 413–418. doi: 10.1016/j.future.2017.09.015

Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing and Management*, *57*(1), 102121. Retrieved from https://doi.org/10.1016/j.ipm.2019.102121 doi: 10.1016/j.ipm.2019.102121

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021, oct). A survey on missing data in machine learning. *Journal of Big Data 2021 8:1*, *8*(1), 1–37. Retrieved from https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00516-9 doi: 10.1186/S40537-021-00516-9

Enamul Haque, M., Tozal, M. E., & Islam, A. (2018). Helpfulness prediction of online product reviews. *Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018*. doi: 10.1145/3209280.3229105

Eslami, S. P., Ghasemaghaei, M., & Hassanein, K. (2018, sep). Which online reviews do consumers find most helpful? A multi-method investigation. *Decision Support Systems*, *113*, 32–42. doi: 10.1016/J.DSS.2018.06.012

Fan, M., Feng, C., Guo, L., Sun, M., & Li, P. (2019). Product-aware helpfulness prediction of online reviews. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, *2*(Ccl), 2715–2721. doi: 10.1145/3308558.3313523

Feng, C., Liang, J., Song, P., & Wang, Z. (2020). A fusion collaborative filtering method for sparse data in recommender systems. *Information Sciences*, *521*, 365–379. Retrieved from `https://doi.org/10.1016/j.ins.2020.02.052` doi: 10.1016/j.ins.2020.02.052

Fkih, F. (2021). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *Journal of King Saud University - Computer and Information Sciences*(September). Retrieved from `https://doi.org/10.1016/j.jksuci.2021.09.014` doi: 10.1016/j.jksuci.2021.09.014

Fkih, F. (2022). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *Journal of King Saud University - Computer and Information Sciences*, *34*(9), 7645–7669. Retrieved from `https://doi.org/10.1016/j.jksuci.2021.09.014` doi: 10.1016/j.jksuci.2021.09.014

Ge, S., Qi, T., Wu, C., Wu, F., Xie, X., & Huang, Y. (2019, dec). Helpfulness-aware review based neural recommendation. *CCF Transactions on Pervasive Computing and Interaction*, *1*(4), 285–295. Retrieved from `https://doi.org/10.1007/s42486-019-00023-0http://link.springer.com/10.1007/s42486-019-00023-0` doi: 10.1007/s42486-019-00023-0

Gupta, V. K., & Rana, P. S. (2020). Ensemble Technique for Toxicity Prediction of Small Drug Molecules of the Antioxidant Response Element Signalling Pathway. *The Computer Journal*. doi: 10.1093/comjnl/bxaa001

Han, J. (2014). *Data mining concepts and techniques*. doi: 10.1109/ICMIRA.2013.45

Hartmann, J., Huppertz, J., Schamp, C., & Heitmann, M. (2019). Comparing automated text classification methods. *International Journal of Research in Marketing*, *36*(1), 20–38. doi: 10.1016/j.ijresmar.2018.09.009

Hercigt, T., Brychcín, T., Svobodat, L., & Konkolt, M. (2016). SemEval-2016 task 5: Aspect based sentiment analysis. *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, 342–349. doi: 10.18653/v1/s16-1055

Hernández-Rubio, M., Cantador, I., & Bellogín, A. (2019). A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction*, *29*(2), 381–441.

Hong, H., Xu, D., Wang, G. A., & Fan, W. (2017). Understanding the determinants of online review helpfulness: A meta-analytic investigation. *Decision Support Systems*, *102*, 1–11. doi: 10.1016/j.dss.2017.06.007

Hou, Y., Yang, N., Wu, Y., & Yu, P. S. (2019). Explainable recommendation with fusion of aspect information. *World Wide Web*, *22*(1), 221–240. doi: 10.1007/s11280-018-0558-1

Hug, N. (2020, aug). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. doi: 10.21105/JOSS.02174

Hutto, C., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international aaai conference on web and social media* (Vol. 8, pp. 216–225).

Idrissi, N., & Zellou, A. (2020). A systematic literature review of sparsity issues in recommender systems. *Social Network Analysis and Mining*, *10*(1). Retrieved from `https://doi.org/10.1007/s13278-020-0626-2` doi: 10.1007/s13278-020-0626-2

Imtiaz, M. N., & Islam, M. K. B. (2020). Identifying Significance of Product Features on Customer Satisfaction Recognizing Public Sentiment Polarity: Analysis of Smart Phone Industry Using Machine-Learning Approaches. *Applied Artificial Intelligence*, *00*(00), 1–17. Retrieved from `https://doi.org/10.1080/08839514.2020.1787676` doi: 10.1080/08839514.2020.1787676

India Today. (2020). *Narendra Modi Speech: PM extends lockdown, Congress asks what about relief for poor.* Retrieved from `https://www.indiatoday.in/india/story/pm-narendra-modi-announcement-on-india-lockdown-2-coronavirus-modi-speech-live-updates-1666672-2020-04-14`

Jain, B. (2020, may). *Lockdown 3.0: Curbs extended for 2 weeks, with some easing.* Retrieved from `https://timesofindia.indiatimes.com/india/lockdown-extension-in-india-curbs-extended-for-2-weeks-with-some-easing/articleshow/75498388.cms`

Jalili, M., Ahmadian, S., Izadi, M., Moradi, P., & Salehi, M. (2018). Evaluating collaborative filtering recommender algorithms: a survey. *IEEE access*, *6*, 74003–74024.

Jiang, C., Duan, R., Jain, H. K., Liu, S., & Liang, K. (2015). Hybrid collaborative filtering for high-involvement products: A solution to opinion sparsity and dynamics. *Decision Support Systems*, *79*, 195–208. Retrieved from http://dx.doi.org/10.1016/j.dss.2015.09.002 doi: 10.1016/j.dss.2015.09.002

Jockers, M. (2020). *Introduction to the Syuzhet package.*

Kabir, M., Kabir, M. M. J., Xu, S., & Badhon, B. (2019). An empirical research on sentiment analysis using machine learning approaches. *International Journal of Computers and Applications*, *0*(0), 1–9. Retrieved from https://doi.org/1206212X.2019.1643584 doi: 10.1080/1206212x.2019.1643584

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, *52*(1), 273–292. Retrieved from https://doi.org/10.1007/s10462-018-09677-1 doi: 10.1007/s10462-018-09677-1

Kamath, S., Kamath, R., & Salins, P. (2020). COVID-19 pandemic in India: challenges and silver linings. *Postgraduate Medical Journal*, postgradmedj–2020–137780. doi: 10.1136/postgradmedj-2020-137780

Kaur, H., & Mangat, V. (2017). A Survey of Sentiment Analysis techniques. In *International conference on i-smac (iot in social, mobile, analytics and cloud) (i-smac 2017)* (pp. 921–925).

Khan, Z. Y., Niu, Z., Sandiwarno, S., & Prince, R. (2021, jan). Deep learning techniques for rating prediction: a survey of the state-of-the-art. *Artificial Intelligence Review*, *54*(1), 95–135. Retrieved from https://doi.org/10.1007/s10462-020-09892-9 doi: 10.1007/s10462-020-09892-9

Kiritchenko, S., Zhu, X., Cherry, C., & Mohammad, S. (2015). NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. (SemEval), 437–442. doi: 10.3115/v1/s14-2076

Kong, L., Li, C., Ge, J., Ng, V., & Luo, B. (2022). Predicting Product Review Helpfulness - A Hybrid Method. *IEEE Transactions on Services Computing*, *15*(4), 2213–2225. doi: 10.1109/TSC.2020.3041095

Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., & Alsaadi, F. E. (2020). Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing Journal*, *86*, 105836. Retrieved from https://doi.org/10.1016/j.asoc.2019.105836 doi: 10.1016/j.asoc.2019.105836

Krishnamoorthy, S. (2015, may). Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, *42*(7), 3751–3759. doi: 10.1016/J.ESWA.2014.12.044

Kumar, P., Kumar, V., & Thakur, R. S. (2019). A new approach for rating prediction system using collaborative filtering. *Iran Journal of Computer Science*, *2*(2), 81–87. Retrieved from https://doi.org/10.1007/

s42044-018-00028-5 doi: 10.1007/s42044-018-00028-5

Li, L., Yuan, H., Qian, Y., & Shao, P. (2018). Towards exploring when and what people reviewed for their online shopping experiences. *Journal of Systems Science and Systems Engineering*, *27*(3), 367–393. doi: 10.1007/s11518-016-5318-0

Lin, K. P., Shen, C. Y., Chang, T. L., & Chang, T. M. (2017). A Consumer Review-Driven Recommender Service for Web E-Commerce. In *2017 ieee 10th conference on service-oriented computing and applications (soca)* (pp. 206–210). Kanazawa, Japan. doi: 10.1109/SOCA.2017.35

Liu, L., Mehandjiev, N., & Xu, D. L. (2011). Multi-criteria service recommendation based on user criteria preferences. *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems*, 77–84. doi: 10.1145/2043932.2043950

Luo, Y., & Xu, X. (2019). Predicting the helpfulness of online restaurant reviews using different machine learning algorithms: A case study of yelp. *Sustainability (Switzerland)*, *11*(19). doi: 10.3390/su11195254

Malik, M. (2020, sep). Predicting users' review helpfulness: the role of significant review and reviewer characteristics. *Soft Computing*, *24*(18), 13913–13928. Retrieved from https://doi.org/10.1007/s00500-020-04767-1http://link.springer.com/10.1007/s00500-020-04767-1 doi: 10.1007/s00500-020-04767-1

Malik, M., & Hussain, A. (2018). An analysis of review content and reviewer variables that contribute to review helpfulness. *Information Processing and Management*, *54*(1), 88–104. Retrieved from http://dx.doi.org/10.1016/j.ipm.2017.09.004 doi: 10.1016/j.ipm.2017.09.004

Malik, M., & Hussain, A. (2020). Exploring the influential reviewer, review and product determinants for review helpfulness. *Artificial Intelligence Review*, *53*(1), 407–427. Retrieved from https://doi.org/10.1007/s10462-018-9662-y doi: 10.1007/s10462-018-9662-y

Masks are mandatory for all now. (2020, apr). Retrieved from https://www.thehindu.com/news/cities/chennai/masks-are-mandatory-for-all-now/article31400601.ece

Matsunami, Y., Ueda, M., & Nakajima, S. (2018). How to Find Similar Users in Order to Develop a Cosmetics Recommender System. *Transactions on Engineering Technologies*, 337–350. doi: 10.1007/978-981-10-7488-2_25

Mauro, N., Ardissono, L., & Petrone, G. (2021). *User and item-aware estimation of review helpfulness* (Vol. 58) (No. 1). doi: 10.1016/j.ipm.2020.102434

McAuley, J. (2018). *Amazon Review Data.* Retrieved 2021-05-14, from https://jmcauley.ucsd.edu/data/amazon/

Mewada, A., & Dewang, R. K. (2021). Research on false review detection Methods: A state-of-the-art review. *Journal of King Saud University - Computer and Information Sciences*(xxxx). Retrieved from https://doi.org/10.1016/j.jksuci.2021.07.021 doi: 10.1016/j.jksuci.2021.07.021

Mitra, S., & Jenamani, M. (2021). Helpfulness of online consumer reviews: A multi-perspective approach. *Information Processing and Management*, *58*(3), 102538. Retrieved from https://doi.org/10.1016/j.ipm.2021.102538 doi: 10.1016/j.ipm.2021.102538

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, *29*(3), 436–465.

Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences*.

Mueller, A. (2020). *scikit-learn · PyPI.* Retrieved 2020-09-10, from https://pypi.org/project/scikit-learn/

Musto, C., Gemmis, M. D., & Semeraro, G. (2017). A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users ' Reviews. In *Proceedings of the eleventh acm conference on recommender systems* (pp. 321–325). doi: 10.1145/3109859.3109905

Naserghandi, A., Allameh, S. F., & Saffarpour, R. (2020). All about COVID-19 in brief. *New Microbes and New Infections*, *35*, 100678. Retrieved from https://doi.org/10.1016/j.nmni.2020.100678 doi: 10.1016/j.nmni.2020.100678

Nehete, S. P., & Devane, S. R. (2018). Recommendation Systems - past , present and future. In *2018 eleventh international conference on contemporary computing (ic3)* (pp. 1–7). IEEE. doi: 10.1109/IC3.2018.8530620

Ni, J., Li, J., & McAuley, J. (2020). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 188–197. doi: 10.18653/v1/d19-1018

Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, *92*, 507–520. Retrieved from https://doi.org/10.1016/j.eswa.2017.09.058 doi: 10.1016/j.eswa.2017.09.058

Noh, Y., Park, S., & Park, S. B. (2019). Aspect-based sentiment analysis using aspect map. *Applied Sciences (Switzerland)*, *9*(16), 1–16. doi: 10.3390/app9163239

Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P.

(2022). Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems*, *64*(1), 35–74.

Patel, B., Desai, P., & Panchal, U. (2017). Methods of recommender system: A review. In *2017 international conference on innovations in information, embedded and communication systems (iciiecs)* (pp. 1–4).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Perrot, M. (2011). Scikit-learn. *Journal of Machine Learning Research*, *12*, 2825–2830. doi: 10.5555/1953048.2078195

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2015). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th international workshop on semantic evaluation (semeval 2014)* (pp. 27–35). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `http://aclweb.org/anthology/S14-2004` doi: 10.3115/v1/s14-2004

Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, *108*, 42–49. doi: 10.1016/j.knosys.2016.06.009

Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, *97*, 205–227. Retrieved from `https://doi.org/10.1016/j.eswa.2017.12.020` doi: 10.1016/j.eswa.2017.12.020

Prabhu, A. N., Kamath, G. B., & Pai, D. V. (2020). Keeping the country positive during the COVID 19 pandemic : Evidence from India. *Asian Journal of Psychiatry*, *51*, 102118. Retrieved from `https://doi.org/10.1016/j.ajp.2020.102118` doi: 10.1016/j.ajp.2020.102118

Rahman, M. A., & Dey, E. K. (2018, may). Datasets for Aspect-Based Sentiment Analysis in Bangla and Its Baseline Evaluation. *Data*, *3*(2), 15. Retrieved from `http://www.mdpi.com/2306-5729/3/2/15` doi: 10.3390/data3020015

Raza, M., Hussain, F. K., Hussain, O. K., Zhao, M., & ur Rehman, Z. (2019). A comparative analysis of machine learning models for quality pillar assessment of SaaS services by multi-class text classification of users' reviews. *Future Generation Computer Systems*, *101*, 341–371. Retrieved from `https://doi.org/10.1016/j.future.2019.06.022` doi: 10.1016/j.future.2019.06.022

Ricci, F., Rokach, L., Shapira, B., Kantor, P. B., & Ricci, F. (2011). *Recommender Systems Handbook*. doi: 10.1007/978-0-387-85820-3

Rodrigues, A. P., Chiplunkar, N. N., & Fernandes, R. (2020). Aspect-based classification of product reviews using Hadoop framework. *Cogent Engineering*, *7*(1). Retrieved from https://doi.org/10.1080/23311916.2020.1810862 doi: 10.1080/23311916.2020.1810862

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1). Retrieved from https://doi.org/10.1186/s40537-022-00592-5 doi: 10.1186/s40537-022-00592-5

Sabeeh, A., & Dewang, R. K. (2019). Comparison, classification and survey of aspect based sentiment analysis. In *Communications in computer and information science* (Vol. 955, pp. 612–629). doi: 10.1007/978-981-13-3140-4_55

Sachan, A., & Richhariya, V. (2013). Reduction of Data Sparsity in Collaborative Filtering based on Fuzzy Inference Rules. *International Journal of Advanced Computer Research*, *3*(2).

Salehan, M., & Kim, D. J. (2016, jan). Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. *Decision Support Systems*, *81*, 30–40. doi: 10.1016/J.DSS.2015.10.006

Saranya, K. G., Sudha Sadasivam, G., & Chandralekha, M. (2016). Performance comparison of different similarity measures for collaborative filtering technique. *Indian Journal of Science and Technology*, *9*(29). doi: 10.17485/ijst/2016/v9i29/91060

Saumya, S., Singh, J. P., Baabdullah, A. M., Rana, N. P., & Dwivedi, Y. K. (2018). Ranking online consumer reviews. *Electronic Commerce Research and Applications*, *29*, 78–89. Retrieved from https://doi.org/10.1016/j.elerap.2018.03.008 doi: 10.1016/j.elerap.2018.03.008

Saumya, S., Singh, J. P., & Dwivedi, Y. K. (2020). Predicting the helpfulness score of online reviews using convolutional neural network. *Soft Computing*, *24*(15), 10989–11005. Retrieved from https://doi.org/10.1007/s00500-019-03851-5 doi: 10.1007/s00500-019-03851-5

Sedhain, S., Menony, A. K., Sannery, S., & Xie, L. (2015). AutoRec: Autoencoders meet collaborative filtering. *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, 111–112. doi: 10.1145/2740908.2742726

Shaheen, M. (2019). Sentiment Analysis on Mobile Phone Reviews Using Supervised Learning Techniques. *International Journal of Modern Education and Computer Science*, *11*(7), 32–43. doi: 10.5815/ijmecs.2019.07.04

Sharma, D., Kumar, B., & Chand, S. (2017). A Survey on Journey of Topic Modeling Techniques from SVD to Deep Learning. *International Journal of Modern Education and Computer Science*, *9*(7), 50–62. doi: 10.5815/ijmecs.2017.07.06

Shen, R.-P., Zhang, H.-R., Yu, H., & Min, F. (2019). Sentiment based matrix factorization with reliability for recommendation. *Expert Systems with Applications*, *135*, 249–258. Retrieved from https://doi.org/10.1016/j.eswa.2019.06.001 doi: 10.1016/j.eswa.2019.06.001

Sidhu, R. K., Kumar, R., & Rana, P. S. (2020). Machine learning based crop water demand forecasting using minimum climatological data. *Multimedia Tools and Applications*, *79*(19-20), 13109–13124. doi: 10.1007/s11042-019-08533-w

Singh, M. (2020). Scalability and sparsity issues in recommender datasets: a survey. *Knowledge and Information Systems*, *62*(1), 1–43. Retrieved from https://doi.org/10.1007/s10115-018-1254-2 doi: 10.1007/s10115-018-1254-2

Sinha, B. B., & Dhanalakshmi, R. (2022). Evolution of recommender paradigm optimization over time. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1047–1059.

Sohail, S. S., Siddiqui, J., & Ali, R. (2017). Classifications of Recommender Systems: A review. *Journal of Engineering Science and Technology Review*, *10*(4), 132–153. doi: 10.25103/jestr.104.18

Sun, X., Han, M., & Feng, J. (2019, sep). Helpfulness of online reviews: Examining review informativeness and classification thresholds by search products and experience products. *Decision Support Systems*, *124*, 113099. doi: 10.1016/J.DSS.2019.113099

Sundar, A. P., Li, F., Zou, X., Gao, T., & Russomanno, E. D. (2020). Understanding shilling attacks and their detection traits: A comprehensive survey. *IEEE Access*, *8*, 171703–171715. doi: 10.1109/ACCESS.2020.3022962

Symeonidis, S., Effrosynidis, D., & Arampatzis, A. (2018). A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications*, *110*, 298–310. Retrieved from https://doi.org/10.1016/j.eswa.2018.06.022 doi: 10.1016/j.eswa.2018.06.022

Taghavi, M., Bentahar, J., Bakhtiyari, K., & Hanachi, C. (2018, mar). New Insights Towards Developing Recommender Systems. *The Computer Journal*, *61*(3), 319–348. Retrieved from https://academic.oup.com/comjnl/article/61/3/319/3893562 doi: 10.1093/COMJNL/BXX056

Tahmasebi, M., Ghazvini, F. F., & Esmaeili, M. (2018). Implementation and evaluation of a resource-based learning recommender based on learning style and web page features. *Journal of Web Engineering*, *17*(3-4), 284–304.

Tamchyna, A., Fiala, O., & Veselovská, K. (2015). Czech aspect-based sentiment analysis: A new dataset and preliminary results. *CEUR Workshop Proceedings*, *1422*, 95–99.

Tao, J., & Fang, X. (2020). Toward multi-label sentiment analysis: a transfer learning based approach. *Journal of Big Data*, *7*(1), 1–26. Retrieved from `https://doi.org/10.1186/s40537-019-0278-0` doi: 10.1186/s40537-019-0278-0

Times, T. E. (2020). *FM's COVID stimulus package: Personal finance relief measures.* Retrieved from `https://economictimes.indiatimes.com/wealth/personal-finance-news/fmscovid-stimulus-package-personal-finance-relief-measures/{%}0Avideoshow/75823472.cms?from=mdr`

Vamshi, K. B., Pandey, A. K., & Siva, K. A. (2018). Topic Model Based Opinion Mining and Sentiment Analysis. In *2018 international conference on computer communication and informatics, iccci 2018* (pp. 1–4). Coimbatore, India: IEEE. doi: 10.1109/ICCCI.2018.8441220

Vandic, D., Frasincar, F., & Kaymak, U. (2018). A framework for product description classification in e-commerce. *Journal of Web Engineering*, *17*(1-2), 1–27. doi: 10.26421/JWE17.1-2

Varghese, A., Agyeman-Badu, G., & Cawley, M. (2020). Deep learning in automated text classification: a case study using toxicological abstracts. *Environment Systems and Decisions*(0123456789). Retrieved from `https://doi.org/10.1007/s10669-020-09763-2` doi: 10.1007/s10669-020-09763-2

Venigalla, A. S. M., Chimalakonda, S., & Vagavolu, D. (2020). Mood of India during Covid-19 - An interactive web portal based on emotion analysis of twitter data. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 65–68. doi: 10.1145/3406865.3418567

Verma, V., & Aggarwal, R. K. (2020). A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: empirical and theoretical perspective. *Social Network Analysis and Mining*, *10*(1), 1–16. Retrieved from `https://doi.org/10.1007/s13278-020-00660-9` doi: 10.1007/s13278-020-00660-9

Wang, Y., Deng, J., Gao, J., & Zhang, P. (2017). A hybrid user similarity model for collaborative filtering. *Information Sciences*, *418-419*, 102–118. Retrieved from `http://dx.doi.org/10.1016/j.ins.2017.08.008` doi: 10.1016/j.ins.2017.08.008

Wen, H., Ding, G., Liu, C., & Wang, J. (2014). Matrix factorization meets cosine similarity: Addressing sparsity problem in collaborative filtering recommender system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8709 LNCS*, 306–317. doi: 10.1007/978-3-319-11116-2_27

Wu, J. (2017). Review popularity and review helpfulness: A model for user review effectiveness. *Decision Support Systems*, *97*, 92–103. Retrieved from http://dx.doi.org/10.1016/j.dss.2017.03.008 doi: 10.1016/j.dss.2017.03.008

Xu, X., Dutta, K., & Ge, C. (2018). Do adjective features from user reviews address sparsity and transparency in recommender systems? *Electronic Commerce Research and Applications*, *29*(October 2017), 113–123. Retrieved from https://doi.org/10.1016/j.elerap.2018.04.002 doi: 10.1016/j.elerap.2018.04.002

Yamin, M. (2020, may). Counting the cost of COVID-19. *International Journal of Information Technology*, *13*(5). Retrieved from https://doi.org/10.1007/s41870-020-00466-0http://link.springer.com/10.1007/s41870-020-00466-0 doi: 10.1007/s41870-020-00466-0

Yang, X., Zhou, S., & Cao, M. (2020). An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews. *Mobile Networks and Applications*, *25*(2), 376–390. doi: 10.1007/s11036-019-01246-2

Yun, Y., Hooshyar, D., Jo, J., & Lim, H. (2017). Developing a hybrid collaborative filtering recommendation system with opinion mining on purchase review. *Journal of Information Science*. doi: 10.1177/0165551517692955

Zhang, G., Liu, Y., & Jin, X. (2020). A survey of autoencoder-based recommender systems. *Frontiers of Computer Science*, *14*(2), 430–450. doi: 10.1007/s11704-018-8052-6

Zhao, G., Lei, X., Qian, X., Mei, T., & Member, S. (2018). Exploring Users ' Internal Influence from Reviews for Social Recommendation : A Case Study on Yelp. *IEEE Transactions on Multimedia*, *21*(3), 771–781. doi: 10.1109/TMM.2018.2863598

APPENDIX A

E-Commerce: Domain Selection

Hi All, Please fill the following form to help us understand which product is the most bought through e-commerce platforms

* Required

1. Name *

____

2. Age *

____

3. Gender *

Mark only one oval.

- Male
- Female
- Other: _______________________________

4. Do you use e-commerce platforms for shopping like Amazon, Flipkart, Myntra or any other? *

Mark only one oval.

- Yes
- No

5. Which e-commerce platform you use the most for online shopping? *

Mark only one oval.

- Amazon
- Flipkart
- Ajio
- Myntra
- Tata Cliq
- Nykaa

6. Do you prefer buying mobile phones through e-commerce platforms? *

Mark only one oval.

- Yes
- No

7. Which product has got the best discount on e-commerce platform? *

Mark only one oval.

- Mobile phone
- Laptop
- Books
- Apparel
- Cosmetics/Beauty
- Security Appliances
- Home Appliances
- Grocery
- Health
- Sports/Fitness equipments
- Toys/Baby Products
- Pet Products

8. Which product your prefer buying on e-commerce platform? *

Mark only one oval.

- Mobile phone
- Laptop
- Books
- Apparel
- Cosmetics/Beauty
- Security Appliances
- Home Appliances
- Grocery
- Health
- Sports/Fitness equipments
- Toys/Baby Products
- Pet Products

9. What makes online shopping through e-commerce platforms preferable for you? *

Mark only one oval.

- Discounts offered
- Doorstep Delivery
- 24*7 buying
- Return/Replacement/Refund Policy
- User Friendly App/Website
- Ease of Product Access
- Product Variety
- Try n Buy Facility

10. Rate your satisfaction for the recommendations shown to you during online shopping *

Mark only one oval.

- Very satisfied
- Satisfied
- Neutral
- Dissatisfied
- Very dissatisfied

11. Suggest a change in e-commerce platforms *

12. Suggest new functionality to be added in e-commerce platforms? *

Survey responses: The responses to the survey can be accessed through the link:

https://drive.google.com/file/d/1WamFnzCSnntcRxZkW5N85jHqLsHxWdnN/view?usp=share_link

APPENDIX B

Mobile Phone Survey

* Required

1. Your Name

____

2. Please select your age group from the below options *

Mark only one oval.

- 16-20
- 21-25
- 26-30
- 31-35
- 36-40
- 41-45
- 46-50
- 51-55
- 56-60
- Above 60

3. Your Gender *

Mark only one oval.

- Female
- Male
- Prefer not to say

4. Your Profession *

Mark only one oval.

- Student
- Private Sector Employee
- Business
- Public Sector Employee
- Retired
- Other: _______________________________

5. Your Education *

Mark only one oval.

- Pursuing Graduation
- Graduate
- Pursuing Post Graduation
- Post Graduate
- Pursuing PhD
- Doctorate

6. Do you own a phone? *

Mark only one oval.

- Yes
- No

7. Your current phone brand? *

Mark only one oval.

- Apple
- Motorola
- Nokia
- Samsung
- Redmi
- Realme
- Vivo
- Oppo
- Honor
- OnePlus
- Other

8. How long have you been using your current phone? *

Mark only one oval.

- less than a year
- 1-2 years
- 2-3 years
- 3-4 years
- 4 years and above

9. Your level of satisfaction with your current phone? *

Mark only one oval.

- Very satisfied
- Satisfied
- Neutral
- Dissatisfied
- Very dissatisfied

10. How much time do you spend using your phone in a day? *

Mark only one oval.

- less than 2 hours
- 2-4 hours
- 4-6 hours
- 6-8 hours
- 8-10 hours
- More than 10 hours

11. Which Operating System wins anyday? *

Mark only one oval.

- iOS
- Android
- Don't know what an Operating System is?
- Other: ______________________________

12. How much are you willing to pay for a mobile phone? *

Mark only one oval.

- less than 10000
- 10000-20000
- 20000-30000
- 30000-40000
- 40000-50000
- Above 50000

13. Which of the following payment mode do you prefer? *

Mark only one oval.

- Cash
- EMI using debit/credit card
- Payments using Wallets (PayTM/AmazonPay/Bhim UPI etc)

14. What are the three most important features for you in a mobile phone? *

Check all that apply.

- [ ] Accessory
- [ ] Battery
- [ ] Camera
- [ ] Processor
- [ ] Screen/Display
- [ ] Hardware
- [ ] Memory
- [ ] Mobile Design
- [ ] Operating System
- [ ] Price
- [ ] Software
- [ ] Warranty
- [ ] Dust resistance
- [ ] Water resistance

15. Which e-commerce platform you feel is good for purchasing a mobile phone? *

Mark only one oval.

- Amazon
- Flipkart
- Both
- Neither
- Prefer buying from offline store

16. Do you consider a product's review before buying it from any e-commerce platform? *

Mark only one oval.

- Yes
- No
- I find them fake/promotional in nature
- At times

17. Do you provide a product review after you've bought it from any e-commerce platform?

Mark only one oval.

- Yes
- No
- At times

18. What kind of experience you post in your reviews of a product?

Mark only one oval.

- Only the positive ones
- Mostly the negative ones
- Critical reviews containing both positive and negative

19. Which is more dependable when buying a product online? *

Mark only one oval.

- User Ratings
- User Reviews
- Both
- None
- I don't know what they are

20. Which company you think has the best phone in market right now? *

Mark only one oval.

- Apple
- Samsung
- Google
- Motorola
- Xioami
- Realme
- Vivo
- Oppo
- Honor
- OnePlus
- Other

21. Any suggestions/improvements for this survey?

Survey responses: The responses to the survey can be accessed through the link:

`https://docs.google.com/spreadsheets/d/1tMKT_SguOw0kUJoKzi8PeK3dW6lO67Uh/edit?usp=share_link&ouid=101973378174219885901&rtpof=true&sd=true`

APPENDIX C

Manually annotated dataset consisting of aspect categories and sentiment can be accessed through the following link: `https://docs.google.com/spreadsheets/d/1f5_vpn4dNE5QDMnzBzruJ5AiJ3mtBK1O/edit?usp=sharing&ouid=101973378174219885901&rtpof=true&sd=true`

BRIEF PROFILE OF RESEARCH SCHOLAR

Dimple Chehal did B.Tech (Computer Science Engineering) in 2013 from Lingaya's University, Faridabad, Haryana, India and M.Tech (Computer Science and Engineering) in 2015 from Maharishi Dayanand University, Rohtak, Haryana, India. She worked as System Engineer for two years at Tata Consultancy Services Pvt. Ltd., Gurugram, Haryana, India. She received fellowship for pursuing PhD from University Grants Commission (UGC), New Delhi, India under Junior Research Fellowship scheme vide letter no. F.15-9(JUNE 2015)/2015(NET). Presently, she is serving as an Assistant Professor in the Department of Computer Engineering, at J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, India.

LIST OF PUBLICATIONS OUT OF THESIS

List of Published Papers

S-No	Title of paper	Journal/ Conference Name	ISSN No.	Vol & Issue	Year	Page
1	Evaluating Annotated Dataset Of Customer Reviews For Aspect Based Sentiment Analysis https://doi.org/10.13052/jwe1540-9589.2122	Journal of Web Engineering, River Publisher	1544-5976 (O), 1540-9589 (P)	21,2	2021	145-178
2	Covid-19 Pandemic Lockdown: An Emotional Health Perspective Of Indians On Twitter https://doi.org/10.1177/0020764020940741	International Journal of Social Psychiatry, Sage Publication	0020-7640, 1741-2854 (O)	67,1	2020	64-72
3	Predicting the Usefulness of E-Commerce Products' Reviews using Machine Learning Techniques https://doi.org/10.31449/inf.v47i2.4155	Informatica, Slovensko Drustvo Informatika	0350-5596, 1854-3871 (O)	47,2	2023	275-284
4	Comparative Study of Missing Value Imputation Techniques on E-Commerce Product Ratings https://doi.org/10.31449/inf.v47i3.4156	Informatica, Slovensko Drustvo Informatika	0350-5596, 1854-3871 (O)	47,3	2023	373-382
5	A New Approach To Identify Similar Users Based On Customer Reviews https://doi.org/10.1088/1757-899X/804/1/012047	IOP Conference Series: Materials Science And Engineering,International Symposium on Fusion of Science and Technology (ISFT 2020), Jan 6-10, 2020, Faridabad, India	1757-899X (O), 1757-8981 (P)	804	2020	012047
6	An Approach to Utilize E-Commerce Product Reviews to Remove Irrelevant Recommendations https://doi.org/10.1109/DELCON54057.2022.9753277	IEEE Delhi International Conference on Electrical, Electronics and Computer Engineering (DELCON-2022), 11-13 Feb, 2022, Delhi, India	-	-	2022	1-3

List of Communicated Papers

SNo	Title of paper	Name of Journal	Present Status	Year
1	ABCF:An Approach to Mitigate Customer Reviews Subjectivity based Sparsity in Recommender Systems	International Journal on Artificial Intelligence Tools	Communicated	2023