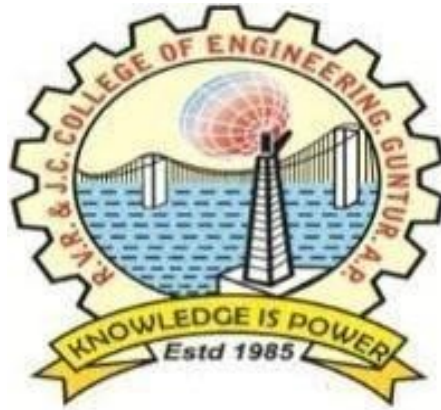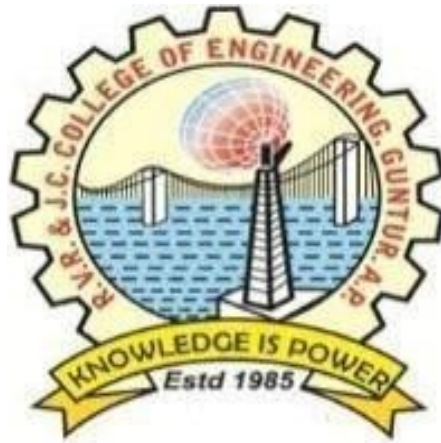**A TERM PAPER REPORT**

ON

PRUS: Product Recommender System Based on

User Specifications and Customer Reviews

Submitted in the partial fulfillment of requirements to

CS- 363 - Term Paper

By

Batch No. 19

ORUGANTI MONIK PAPARAO(Y22CS139)

PENDYALA SKANDA BHAGAVAN (Y22CS145)

TULAM SAI SUDHEER (Y22CS184)

R.V.R. & J.C. COLLEGE OF ENGINEERING (Autonomous)

(NAAC 'A+' Grade)

Approved by AICTE :: Affiliated to Acharya Nagarjuna University

Chandramoulipuram::Chowdavaram

GUNTUR – 522 019,Andhra Pradesh, India.

R.V.R & J.C. COLLEGE OF ENGINEERING(Autonomous)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is to certify that this Term Paper titled "PRUS: Product Recommender System Based on User Specificationand Customer Reviews" is the study conducted by **ORUGANTI MONIK PAPARAO (Y22CS139), PENDYALA SKANDA BHAGAVAN (Y22CS145), TULAM SAI SUDHEER (Y22CS184)** submitted in partial fulfillment of the requirements to CS 363- Term Paper during the Academic Year 2024-2025.

Smt. N. Zareena   **Dr. S J R K Padminivalli V**   **Dr. M. Sreelatha**

Term Paper Guide   Term Paper In- Charge   Prof. &Head,CSE

ACKNOWLEDGEMENT

ORUGANTI MONIK PAPARAO (Y22CS139)

PENDYALA SKANDA BHAGAVAN (Y22CS145)

TULAM SAI SUDHEER (Y22CS184)

ABSTRACT

The growing popularity of online shopping has led to a continuous influx of product reviews, which play a crucial role in shaping consumer purchasing decisions. Many paper studies have focused on ranking products using these reviews. However, most methodologies tend to overlook the impact of negative sentiments when evaluating products based on customer preferences. This paper aims to address this gap by incorporating both positive and negative polarity in product ranking. To achieve this, the proposed method first breaks down reviews into individual sentences, allowing for a more granular analysis of sentiment at the phrase level. By extracting key elements from these reviews, the approach effectively determines the polarity of different product features. The next step involves linking these polarities to sentence-level attributes within the review. To ensure that product rankings align with user preferences, relative importance is assigned to each polarity based on specific needs. The effectiveness of this approach has been tested using the Amazon review dataset, where experimental evaluations were conducted using rank score (RS) and normalized discounted cumulative gain (nDCG) score. Results demonstrate that the proposed ranking system, PRUS, enables users to customize their recommended product lists by prioritizing either positive or negative aspects based on their preferences.

Contents   Page. No.

LIST OF TABLES

LISTOF FIGURES

LIST OF ABBREVIATIONS

PRUS	-	Product Recommender System Based on User Specification
AAS	-	Average Aspect Score
RS	-	Rank Score
QAM	-	Query Aspect Modeling
NLP	-	Natural Language Processing
SVM	-	Super Vector Machine
GP	-	Genetic Programming
NLTK	-	Natural Language Toolkit
DCG	-	Discounted Cumulative Gain
nDCG	-	Normalized Discounted Cumulative Gain
IDCG	-	Ideal Discounted Cumulative Gain

1. INTRODUCTION

1.1 Background

The rapid expansion of the e-commerce industry has significantly transformed the way people shop, introducing an unparalleled level of convenience and accessibility. In the past, shopping was largely an in-person experience, requiring consumers to visit physical stores, compare products manually, and rely on sales representatives for information. However, with the proliferation of e-commerce platforms, consumers can now browse through thousands of products, compare prices, and make purchases from the comfort of their homes. Online marketplaces such as Amazon, eBay, Alibaba, Flipkart, and Walmart have made it easier than ever to access a diverse range of products, often at competitive prices. The convenience of being able to shop anytime and from anywhere has made online shopping the preferred choice for millions of people around the world. Additionally, the availability of mobile apps, secure payment gateways, and faster delivery services has further accelerated the shift from traditional retail to digital shopping.

A key factor driving the growth of e-commerce is the ability to access detailed product information before making a purchase. Online platforms provide comprehensive product descriptions, specifications, and promotional details, allowing consumers to make informed choices. More importantly, customers can read user-generated reviews that offer honest feedback based on personal experiences. These reviews have become an integral part of the decision-making process, as they often reveal insights that cannot be gleaned from product descriptions alone. For instance, while a product listing may highlight the technical features of a smartphone, customer reviews provide practical insights into its real-world performance, such as battery life, camera quality, and durability. This additional layer of information allows potential buyers to make more confident purchasing decisions. One of the most significant outcomes of the e-commerce boom is the proliferation of customer reviews across various platforms. Consumers frequently share their experiences by posting feedback on e-commerce websites, social media platforms, and specialized review sites. This trend has given rise to a review-driven shopping culture, where buyers heavily rely on the opinions of previous customers. Reviews often detail both the positive and negative aspects of products, offering a balanced perspective that helps other shoppers evaluate the pros and cons. For example, a review of a laptop might praise its fast processing speed and sleek design but criticize its short battery life.

Similarly, a review of a vacuum cleaner could highlight its powerful suction but mention excessive noise or durability issues. These detailed, real-world experiences

provide prospective buyers with valuable insights that marketing content alone cannot offer.

The influence of customer reviews extends beyond individual product assessments—it also plays a crucial role in shaping brand reputation and consumer trust. Positive reviews can enhance a product's credibility and encourage more sales, while negative reviews can deter potential buyers. E-commerce platforms often feature review aggregation systems, which display average ratings and review scores, giving consumers a quick overview of a product's performance. Products with higher ratings and positive feedback tend to enjoy greater visibility and higher sales, while those with consistently negative reviews may struggle to attract buyers. As a result, businesses are increasingly recognizing the power of customer feedback and actively seeking to improve their products and services based on review insights.

Furthermore, the detailed nature of online reviews has made them a valuable resource for identifying product strengths and weaknesses. Many reviews describe specific aspects of products, allowing potential buyers to focus on the features that matter most to them. For instance, when purchasing a smartphone, some consumers may prioritize camera quality, while others may care more about battery life or display resolution. By reading detailed reviews, shoppers can identify whether a product meets their specific needs, enhancing their overall satisfaction with the purchase. This aspect-level evaluation has become particularly valuable in categories such as electronics, home appliances, and fashion, where individual features significantly impact the user experience.

In addition to influencing purchasing decisions, customer reviews also play a role in driving product improvements and innovation. Many companies actively monitor customer feedback to identify recurring issues, gather suggestions, and make necessary enhancements. For example, if multiple customers highlight poor battery performance in a new smartphone model, the manufacturer may address this in subsequent versions. Similarly, consistent praise for a particular feature may encourage companies to prioritize it in future product iterations. This feedback loop allows businesses to respond to consumer needs more effectively, fostering brand loyalty and customer satisfaction.

The review culture in e-commerce has also introduced new challenges, such as fake reviews and opinion spam. To influence product ratings, some businesses or third parties post fraudulent positive reviews to boost sales or negative reviews to discredit competitors.

This unethical practice can mislead consumers and undermine trust in the review system. To combat this issue, many e-commerce platforms have implemented review

verification processes and AI-powered algorithms to detect suspicious or fake reviews. Verified purchase labels, review authenticity indicators, and reporting mechanisms help maintain the credibility of online reviews, ensuring that customers receive reliable information.

Ultimately, the e-commerce industry has revolutionized the way people shop, and customer reviews have become a cornerstone of the online shopping experience. They empower consumers with real-world insights, shape brand perceptions, and drive purchasing decisions. As e-commerce continues to evolve, the role of customer reviews will only grow stronger, influencing product development, marketing strategies, and consumer trust in the digital marketplace.

Fig-1.1.1

Customer reviews not only benefit potential buyers but also offer valuable data for companies. Businesses use this feedback to identify product strengths and weaknesses, improve existing products or develop new ones based on customer preferences, and enhance customer service by addressing common complaints or issues.

While customer reviews provide valuable insights, the sheer volume of online reviews makes it difficult to extract meaningful information efficiently. As e-commerce platforms grow, the number of reviews for popular products can reach into the thousands, making manual analysis impractical.

1.2 Problem Statement

While customer reviews are a valuable source of information for both consumers and businesses, they often present a challenge due to the mixed sentiments they contain. Many reviews express contrasting opinions about different product features, but most existing product ranking methods fail to capture this complexity. Traditional ranking systems typically focus on the overall sentiment polarity of the review—classifying it as either positive, negative, or neutral—without considering the individual aspects being praised or criticized. This one-dimensional evaluation can lead to misleading conclusions, as it overlooks the nuanced feedback regarding specific product features. For instance, a review for a smartphone may commend its camera quality and battery life but criticize its screen resolution. Despite the negative feedback on the screen, the review may still be classified as positive overall, creating a false impression of the product's suitability for buyers who prioritize display quality.

The oversimplification of review sentiment by current ranking models poses a significant issue for consumers seeking feature-specific information. Since most ranking methods aggregate overall review polarity, they fail to highlight feature-level strengths and weaknesses, making it difficult for users to identify products that truly match their preferences. For example, a smartphone with a top-tier camera but subpar battery life may receive an overall high rating due to the predominance of positive camera-related reviews. However, a buyer who values long battery performance over camera quality might find this phone unsatisfactory. The lack of aspect-level differentiation means that such a buyer could be misled by the aggregate rating, purchasing a product that does not meet their primary needs.

Moreover, customer reviews often contain contradictory sentiments within the same review, further complicating the reliability of overall sentiment-based product rankings. For example, a review of a laptop might praise its build quality and design while simultaneously criticizing its customer support service. If the ranking algorithm considers only the overall sentiment score, it may fail to reflect the negative experience with customer support. This flattening of contrasting opinions into a single sentiment score reduces the informative value of the review. Potential buyers, particularly those who consider customer service a key factor, might not be able to discern this important detail, resulting in less informed purchasing decisions.

To overcome these limitations, aspect-based sentiment analysis (ABSA) offers a more refined and accurate approach to review evaluation. ABSA focuses on identifying and extracting sentiments related to specific product features, providing a detailed

breakdown of customer opinions. Unlike traditional methods that classify the review as a whole, ABSA analyzes the text at the aspect level, allowing for fine-grained sentiment categorization. For example, in a review for a wireless headset, ABSA can separately assess the sentiments regarding sound quality, battery life, comfort, and connectivity stability. This allows potential buyers to filter products based on the features that matter most to them, making the review information far more useful and actionable.

The implementation of ABSA in e-commerce platforms could significantly enhance product recommendations. By associating each product with feature-specific sentiment scores, platforms can enable users to search for products according to specific priorities. For instance, a buyer looking for a laptop with excellent battery life can prioritize reviews specifically highlighting battery performance, rather than relying on the overall sentiment score. This level of specificity improves the accuracy of product recommendations, making it easier for buyers to find products that align with their preferences.

Additionally, ABSA can help e-commerce platforms generate detailed review summaries that highlight both the positive and negative aspects of a product. Instead of displaying a single overall rating, platforms could present aspect-specific ratings (e.g., 4.5/5 for camera quality, 3.2/5 for battery life, and 4.0/5 for design), giving potential buyers a more comprehensive view of the product's strengths and weaknesses. This transparency would allow consumers to make better-informed decisions, reducing the likelihood of post-purchase dissatisfaction.

Furthermore, businesses can leverage ABSA to gain more precise insights into customer feedback. By identifying frequently mentioned positive and negative features, companies can prioritize product improvements more effectively. For example, if ABSA reveals that customers frequently complain about a laptop's overheating issues, the manufacturer can focus on enhancing the cooling system in the next iteration. Similarly, if customers consistently praise the ease of setup of a smart home device, the company can emphasize this feature in its marketing campaigns. This granular understanding of customer feedback enables businesses to make data-driven decisions, enhancing both product quality and customer satisfaction.

1.3 Objectives

The primary objectives of this paper are:

- **Development of the PRUS framework**: Propose a Product Recommendation based on User Specification (PRUS) framework to generate a customized product list by considering user-specified features and preferences. This framework aims to offer more tailored recommendations by addressing individual needs rather than relying on generalized rankings.

- **Multi-polarity sentiment extraction**: Implement a technique to identify and separate positive and negative sentiments within a single review, ensuring that conflicting opinions are not flattened into a single polarity. This allows the system to capture diverse opinions accurately.

- **Weighted sentiment scoring**: Introduce a technique to assign weights to both positive and negative sentiments of product features. This enables the ranking system to prioritize products based on the user's specific preferences, such as giving more importance to certain features (e.g., battery life) over others.

- **User-defined prioritization**: Allow users to define the importance of different product aspects when generating recommendations. This ensures the final ranked list reflects the user's unique needs and preferences, making the recommendations more relevant and valuable.

- **Customized product ranking**: Implement a dynamic ranking mechanism that reorders products based on the weighted sentiment scores of their aspects. This ensures that products with stronger performance in the features valued by the user are ranked higher.

- **Experimental validation**: Perform extensive evaluations using the Amazon review dataset to validate the effectiveness and accuracy of the PRUS framework. Use performance metrics such as rank score (RS) and normalized discounted cumulative gain (nDCG) to measure the precision and reliability of the system.

- **Improved decision-making**: Enhance the decision-making process for both consumers and businesses by providing more accurate, aspect-based product insights. This will help buyers make better-informed decisions and allow companies to identify product strengths and areas for improvement.

- **Scalability and adaptability**: Ensure that the PRUS framework is scalable and adaptable to different product categories and review datasets.

1.4 Limitations of Existing Systems

Despite the availability of various product ranking systems, several limitations persist in existing approaches:

- **Focus on overall sentiment**: Most existing methods primarily consider the overall sentiment of reviews, overlooking the detailed opinions on individual product features. By only capturing the general polarity of a review, these systems fail to account for feature-specific strengths and weaknesses. This can lead to inaccurate and generalized recommendations, preventing users from identifying products that align with their specific needs.

- **Lack of user-specific customization**: Many systems do not allow users to specify the importance of particular features when generating product recommendations. As a result, all product aspects are treated with equal significance, regardless of the user's priorities. For instance, a customer looking for a laptop with long battery life may receive recommendations that prioritize display quality or performance, which may not match their needs. The absence of user-driven customization reduces the relevance and effectiveness of the recommendations.

- **Inefficient sentiment extraction**: Existing models often struggle to accurately extract and link sentiments to specific product aspects. Sentiments expressed in reviews can be ambiguous or multi-faceted, making it challenging to associate the correct polarity with the appropriate feature. For example, a review mentioning, "*The phone's display is stunning, but the battery drains quickly*" contains contrasting sentiments about different aspects. However, many ranking systems fail to distinguish between them, resulting in less precise and unreliable recommendations.

- **Limited consideration of negative polarity**: Most product ranking systems prioritize positive sentiment while neglecting the significance of negative opinions. This creates a bias toward favourable reviews, leading to misleading recommendations. Negative feedback often highlights critical product flaws, such as poor durability or customer service issues, which are valuable for potential buyers. By underweighting or ignoring negative polarity, existing systems provide incomplete and overly optimistic product evaluations, making it harder for users to make informed decisions.

- **Inability to handle mixed or conflicting sentiments**: Many reviews contain conflicting opinions about different product features, but existing methods often fail to distinguish between them.

- This simplified interpretation reduces the accuracy of the recommendations, as the

nuanced user experience is lost.

- **Failure to capture context-specific sentiments**: Sentiment analysis models often miss contextual nuances, such as sarcasm, irony, or subtle expressions of dissatisfaction. For instance, the sentence "*The product works... when it feels like it*" expresses negative sentiment with sarcasm, but many models misinterpret it as neutral or positive. This leads to inaccurate sentiment classification.

- **Limited adaptability across product categories**: Many existing systems are designed for specific product types, making them less adaptable to other categories. A ranking model tailored for electronics may not effectively capture the features and sentiments relevant to clothing or home appliances, limiting its applicability across diverse e-commerce platforms.

- **Static product ranking models**: Most systems generate static product rankings that do not dynamically update based on evolving user preferences or review trends. As customer opinions and product performance shift over time, static models become outdated, providing irrelevant or obsolete recommendations.

- **Lack of multi-polarity analysis**: Existing models often fail to perform multi-polarity sentiment analysis, where both positive and negative opinions within a single review are evaluated independently. For example, if a review praises a product's design but criticizes its functionality, both polarities should be considered separately. However, many systems merge conflicting sentiments, reducing the granularity and accuracy of the analysis.

- **Insufficient consideration of review credibility**: Current systems do not adequately account for fake or biased reviews, which can distort product rankings. Spam reviews or paid promotions often create misleading sentiments, making the recommendations unreliable.

- **Ineffective performance evaluation metrics**: Many existing models rely on basic evaluation metrics, which do not fully capture the effectiveness and accuracy of the ranking system. As a result, the performance of the models in real-world scenarios may not be properly measured, leading to suboptimal recommendations.

2. LITERATURE REVIEW

2.1 Opinion-based entity ranking

[14] K. Ganesan and C. Zhai, ''Opinion-based entity ranking,'' Inf. Retr., vol. 15, no. 2, pp. 116–150, Apr.

This paper introduces a novel method to rank entities, such as hotels and cars, based on user preferences expressed through keyword queries. Unlike existing methods that focus on summarizing opinions, the proposed approach uses the text from all reviews of an entity to represent it. Given a user's query, which specifies desired features, the method ranks entities by how well the reviews align with the user's preferences. The paper explores several text retrieval models, including both standard and extended versions, to address this task. Experiments are conducted in two domains—hotels and cars—demonstrating that the extended models improve ranking accuracy. The method focuses on directly matching user preferences with the content of reviews, offering a more personalized ranking. This allows users to discover entities that best match their individual needs, rather than relying on general ratings or aggregated opinions. The results show that the extended models outperform standard retrieval models for ranking based on opinionated content. The study highlights the potential of opinion matching for better entity discovery. Ultimately, the paper suggests that leveraging opinionated content in this way can be the most significantly improve rankingprecision and user satisfaction.

2.2 Aspect-based opinion ranking framework forproduct reviews

[17] A. Kumar and S. Abirami, ''Aspect-based opinion ranking framework for product reviews using a Spearman's rank correlation coefficient method,'' Inf. Sci., vols. 460–461, pp. 23–41, Sep.

This paper introduces a new framework for ranking products based on aspects, using opinion mining, or sentiment analysis, to analyze social media content. Opinion mining detects emotions and opinions from structured, semi-structured, and unstructured content at various levels, including document, word, sentence, and aspect levels. The aspect level refers to specific attributes or parts of an entity, highlighting users' likes and dislikes. The proposed system first identifies the aspects of products from reviews. Next, it visualizes the aspects and their associated opinion words using a Harel–Koren fast multiscale layout. A network visualization of these aspects is created, and a Spearman's

9

rank correlation coefficient is used for opinion ranking, sorting products based on positive and negative feedback. The system also employs supervised learning methods like Naïve Bayes, Maximum Entropy, and Support Vector Machine for aspect-based sentiment classification. These models help classify the sentiments towards the identified aspects of products. The effectiveness of the proposed framework is evaluated through experimental results. Ultimately, the paper demonstrates the potential of aspect-based opinion mining for improving product ranking and decision-making.

2.3 Opinion-Based Entity Ranking using learning to rank

[18] S. Bashir, W. Afzal, and A. R. Baig, ''Opinion-based entity ranking using learning to rank,'' Appl. Soft Comput., vol. 38, pp. 151–163, Jan. 2016.

As social media and e-commerce continue to grow, opinions have become a vital source of information for decision-making. However, the overwhelming volume of opinions makes it difficult for individuals to evaluate them effectively. Opinion-Based Entity Ranking (OpER) addresses this challenge by directly ranking entities based on how well opinions about them align with a user's preferences, provided as query keywords. This reduces the need to read countless reviews. Previous OpER models primarily compute entity relevance by matching query keywords across all opinions collectively, without considering the subjectivity or importance of individual opinions. This overlooks the fact that entities with positive judgments and strong keyword relevance should be ranked higher than those with negative or weak relevance. To enhance OpER effectiveness, the paper proposes a new framework that matches individual opinions to query keywords, creating a more intuitive ranking model. The authors use genetic programming (GP) to combine various ranking features into a more effective retrieval model. Experimental results demonstrate that the proposed model significantly outperforms the standard OpER approach.

2.4 A Two-Fold Rule-Based Model for Aspect Extraction

[19] T. A. Rana and Y.-N. Cheah, ''A two-fold rule-based model for aspect extraction,'' Expert Syst. Appl., vol. 89, pp. 273–285, Dec. 2017.

Opinion target extraction plays a vital role in aspect-based sentiment analysis by identifying the specific targets of users' opinions in online reviews. While syntactic patterns-based approaches have shown promising results in recent years, they come with limitations. These methods rely on dependency parsers that generate syntactic relations according to grammatical rules and language structures. However, in real-world scenarios, online reviewers often use informal language, slang, and fragmented sentences, making it difficult for parser-dependent methods to accurately extract aspects. To overcome this challenge, the paper introduces a two-fold rules-based model (TF-RBM) designed to enhance aspect extraction accuracy. The first fold identifies aspects linked to domain-independent opinions, capturing general sentiments across different contexts. The second fold targets domain-dependent opinions, which are specific to particular products or services. Furthermore, the authors incorporate frequency- and similarity-based techniques to refine the extraction process, ensuring more precise and reliable results. Experimental evaluations demonstrate that the proposed model outperforms existing state-of-the-art approaches.

2.5 Aspect Term Extraction and Opinion Target Extraction
[20] C. Wu, F. Wu, S. Wu, Z. Yuan, and Y. Huang, ''A hybrid unsupervised method for aspect term and opinion target extraction,'' Knowl.-Based Syst., vol. 148, pp. 66–73, May

In the field of fine-grained sentiment analysis, Aspect Term Extraction (ATE) and Opinion Target Extraction (OTE) play a crucial role in identifying and analyzing specific components of opinions. ATE focuses on extracting the particular features or attributes of an entity being discussed. For example, in a product review, the sentence *"The battery life is impressive"* contains the aspect term "battery life", as it highlights the specific feature being evaluated. By identifying these aspects, businesses can understand which features customers find satisfactory or problematic, allowing them to make targeted improvements. On the other hand, Opinion Target Extraction (OTE) identifies the targets of opinions, which are the objects or entities receiving the sentiment. For instance, in the sentence *"The camera is fantastic"*, the term "camera" is the opinion target, as it is the object being

praised. OTE is essential for linking opinions to their corresponding targets, enabling more precise sentiment classification. Without OTE, it would be difficult to determine whether a sentiment refers to the entire product or a specific component, leading to inaccurate or vague insights.

2.6 Extracting prominent review aspects from customer feedback
[21] Z. Luo, S. Huang, F. F. Xu, B. Y. Lin, H. Shi, and K. Zhu, ''ExtRA: Extracting prominent review aspects from customer feedback,'' in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 3477–3486

Many existing systems for analyzing and summarizing customer reviews focus on identifying prominent review aspects. Traditionally, these aspects are manually determined, which is both costly and inefficient. This approach cannot scale to large platforms like Amazon, Taobao, or Yelp, where countless product types and new products are frequently introduced. To overcome this challenge, the paper proposes a novel framework called ExtRA for automatically extracting the most prominent aspects from textual reviews. ExtRA identifies the K most significant aspect terms or phrases without supervision, ensuring there is no semantic overlap between the extracted aspects. The framework is designed to be scalable, making it suitable for large and diverse product types. It uses unsupervised methods, eliminating the need for manual effort. Extensive experiments show that ExtRA achieves state-of-the-art performance.

2.7 Association rule mining using fuzzylogic and whale optimization
[22] S. Sharmila and S. Vijayarani, ''Association rule mining using fuzzy logic and whale optimization algorithm,'' Soft Comput., vol. 25, no. 2, pp. 1431–1446, Jan. 2021.

Association Rule Mining (ARM) is a data mining technique used to discover frequently co-occurring itemsets in transactional datasets. It involves two main steps: frequent itemset identification and association rule generation, using minimum support and confidence measures. To handle large datasets, the proposed algorithm applies a dimensionality reduction technique using low variance and hash tables, reducing irrelevant transactions and items. This improves efficiency by minimizing database scans and memory usage. The second step uses fuzzy logic combined with whale optimization for

frequent item identification and rule generation. The algorithm's performance is compared with EFP, particle swarm optimization, and genetic algorithms. Metrics include item and transaction reduction, execution time, and memory usage. Experimental results show that the proposed method outperforms existing techniques. It effectively identifies significant patterns and generates reliable association rules.

2.8 Opinion fraud detection in online reviews by network effects
[15] L. Akoglu, ''Opinion fraud detection in online reviews by network effects,'' in Proc. ICWSM, vol. 7, no. 1, 2013, pp. 2–11.

User-generated online reviews significantly impact the success of products, hotels, and restaurants. However, opinion spammers create fake reviews to distort product quality. FRAUDEAGLE is a fast and effective framework designed to detect fraudsters and fake reviews in online datasets. It leverages network effects among reviewers and products, unlike traditional methods that focus on text or behavior analysis. The framework consists of two steps: scoring users and reviews for fraud detection, and grouping for visualization. It operates in an unsupervised manner, requiring no labeled data but can incorporate side information. FRAUDEAGLE is scalable, with runtime growing linearly with network size. It effectively identifies fraud-bots in large datasets, including online app reviews. The framework proves its efficiency on both synthetic and real datasets. It offers a reliable solution for combating online review fraud.

2.9 A survey on implicit aspect detection for sentiment analysis: Terminology, issues, and scope

[9] P. K. Soni and R. Rambola, ''A survey on implicit aspect detection for sentiment analysis: Terminology, issues, and scope,'' IEEE Access, vol. 10, pp. 63932–63957, 2022

Sentiment analysis is a growing paper field that extracts sentiments from text for analysis and aggregation. Aspect-level sentiment analysis focuses on evaluating sentiment for different aspects of entities. While most paper detects explicit aspects, implicit aspects, hinted at by other words, are often overlooked. Since many sentences contain implicit aspects, their detection is crucial for accurate sentiment analysis. This survey explores various methods for implicit aspect detection, categorizing them by applied algorithms. It provides a quantitative evaluation of different methods for comparison. The study also discusses terminology, challenges, and the scope of detecting implicit aspects. Fine-grained sentiment information collected can benefit multiple applications across domains. The survey highlights the need for improved implicit aspect detection techniques. It also identifies future paper trends and suggests ways to enhance performance.

2.10 Aspectlevel sentiments for calling app recommendation

[10] N. Aslam, K. Xia, F. Rustam, A. Hameed, and I. Ashraf, ''Using aspectlevel sentiments for calling app recommendation with hybrid deep-learning models,'' Appl. Sci., vol. 12, no. 17, p. 8522, Aug. 2022.

The widespread use of mobile phones has driven a surge in mobile app development, especially for calling apps. These apps offer free services like messaging, video calls, and audio messages. Despite their varied features, existing studies often overlook aspect-level analysis. This study evaluates IMO, Skype, Telegram, WeChat, and WhatsApp based on account, app, call, message, update, video, and working features. A large dataset from the Google Play store is used for aspect extraction with the Latent Dirichlet Allocation (LDA) model. The apps are analyzed and recommended based on users' call, message, and video preferences. Sentiment analysis is applied to assess user opinions on the apps. A novel ensemble model combining a gated recurrent unit and convolutional neural network achieves 94% accuracy. The study offers insights into app performance and user satisfaction. It highlights the importance of aspect-level analysis.

3 METHODOLOGIES USED

3.4 Architecture

The PRUS framework (Product Recommendation based on User Specification) is an advanced system designed to deliver personalized product recommendations by leveraging aspect-level sentiment analysis. Unlike traditional product ranking methods that rely on overall review polarity, PRUS focuses on individual product features, ensuring that the recommendations align with user-specified preferences. The framework consists of several key stages, each contributing to the accurate extraction, evaluation, and ranking of products based on their aspect-level sentiments.

The process begins with the data retrieval phase, where the system collects customer reviews from a review dataset. These reviews form the input data for the recommendation system. Instead of analyzing the reviews as a whole, PRUS breaks them down into individual sentences, allowing the system to conduct granular sentiment analysis. This segmentation is crucial because customer reviews often contain mixed opinions. By analyzing sentences individually, the framework ensures that both positive and negative sentiments associated with different features are captured separately, preventing misleading overall sentiment interpretations.

In the feature extraction phase, each sentence is processed to identify specific product features and the corresponding sentiment polarity (positive, negative, or neutral). To enhance the accuracy and relevance of the recommendations, the framework employs information gain (IG) as an evaluation method. Information gain, based on entropy calculation, measures how much predictive value a particular feature contributes to the overall classification. Features with higher information gain are deemed more informative and influential, increasing their importance in the ranking process. For example, in a laptop review, features such as battery performance or processing speed may carry more weight if they are frequently mentioned and have a significant impact on the overall user experience. This weighted evaluation ensures that the most relevant and influential product features are prioritized.

Once the features and their associated sentiments are extracted, the system appends the sentiment polarity to the respective feature. The next step is to match these extracted features with the user-specified preferences. This enables the system to prioritize products that align with the user's desired attributes.
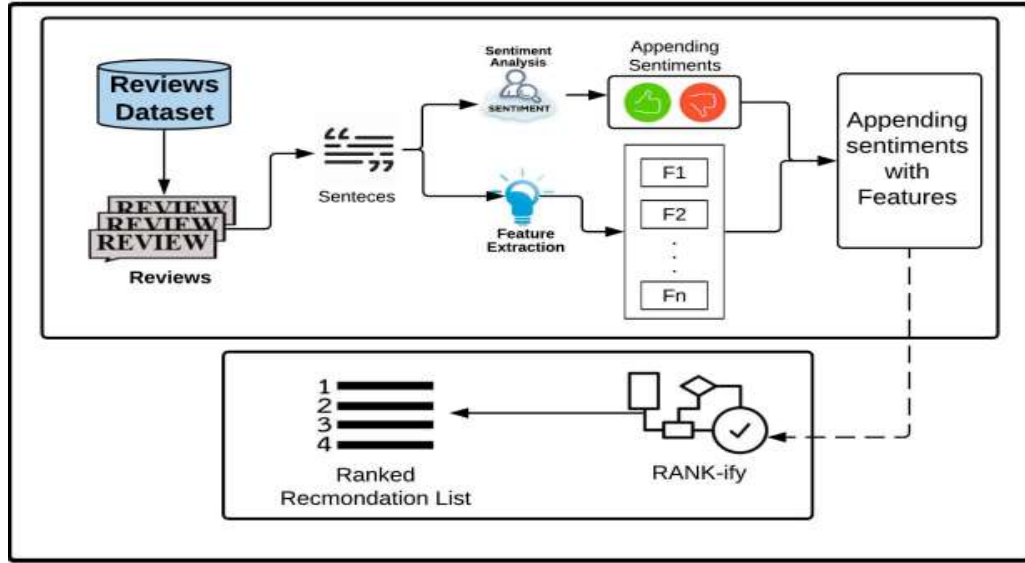
Fig. 3.1.1

The ranking phase is executed using the **RANK-ify algorithm**, which plays a crucial role in assigning weights based on sentiment polarity. The algorithm increases the weight of features with positive sentiments, making them more influential in the final ranking. Conversely, features with negative sentiments are assigned lower weights, decreasing their influence. This weighted sentiment scoring ensures that products with more favourable feedback on the desired features appear higher in the recommendation list. By considering both positive and negative sentiments, the system offers a balanced and realistic evaluation of products, preventing bias towards only positive opinions.

Finally, the ranked recommendation list presents the products along with their features and corresponding sentiment scores, weighted according to the user's preferences. This output provides a personalized, sentiment-driven ranking, making it easier for users to identify products that best match their specific needs. For example, a user seeking a smartphone with a good camera and long battery life would receive a list where products excelling in these areas are ranked higher, regardless of their overall review sentiment.

The PRUS framework offers several key advantages:

- **Granular sentiment analysis:** By breaking reviews into sentences, it captures feature-specific opinions instead of general sentiments.

- **Enhanced precision with information gain:** The use of information gain ensures that the most relevant and informative features carry greater influence.

- **Personalized recommendations:** The framework prioritizes products based on user-defined preferences, ensuring tailored recommendations.

16

3.5 Datasets Used

The dataset used in this study consists of 413,840 reviews, primarily collected through a data crawler from Amazon, specifically focusing on mobile phone reviews from various brands and companies. The dataset contains three key attributes:

- **Product title**: The name of the mobile phone being reviewed.
- **Brand**: The company that manufactures the mobile phone.
- **Review text**: The actual customer feedback or opinion.

3.2.1 Data Preprocessing

To ensure the quality and consistency of the data, a thorough preprocessing phase was conducted. During this step:

- Reviews with missing product titles or brand names were removed,
- Reviews containing fewer than five words were also eliminated, as they were unlikely to offer any meaningful feature or sentiment information.
- The remaining review text underwent cleaning and normalization processes to reduce noise and enhance the quality of the data. This included:
  - Removing stop words and punctuation to eliminate irrelevant terms.
  - Converting integers into words for consistency.
  - Stemming and lemmatizing using the WordNet Lemmatizer, which reduced words to their base or root form. This helped minimize word variations and standardize different writing styles, making the feature extraction process more effective.

Table No. 3.2.1

17

3.2.2 Refined Dataset

After the preprocessing phase, the dataset was refined and reduced to 316,811 reviews, which were used for the experimental evaluation of the proposed PRUS framework. The cleaning process ensured that the dataset contained high-quality, meaningful reviews by eliminating incomplete, short, or irrelevant entries. The reduced dataset offers a substantial and diverse sample for analysis, making the evaluation process more reliable and robust.

The cleaned dataset covers 371 different mobile phone brands, representing a broad spectrum of manufacturers. This diversity allows the framework to be tested across a variety of products, ensuring that the evaluation is not limited to a few popular brands. The inclusion of both major and lesser-known brands adds depth to the dataset, making the analysis more comprehensive.

Upon further examination, it was found that approximately 40% of the reviews were related to just three major brands: Samsung, BLU, and Apple. This indicates the dominance and popularity of these brands in the dataset. Their higher review volume reflects their market influence and customer base, making them significant players in the analysis. The concentration of reviews around these three brands also suggests that they attract more customer attention and engagement, possibly due to their widespread availability, brand reputation, and extensive marketing efforts.

The dataset spans reviews for approximately 4,300 different mobile phone models, showcasing a large and varied product sample. This diversity ensures that the PRUS framework is tested on a wide range of products with different specifications, performance levels, and customer experiences. Analyzing such a large number of models helps the system generalize better, making it applicable to a variety of products and not just a limited subset.

The analysis further revealed that the top ten most reviewed products in the dataset received between 800 and 1000 reviews each, indicating their popularity and high customer engagement. The large volume of reviews for these products highlights their frequent customer feedback, making them ideal candidates for detailed sentiment analysis. The presence of multiple reviews allows the PRUS framework to accurately identify trends, patterns, and recurring sentiments associated with these products. The higher review count also makes the sentiment analysis more reliable, as a larger sample size reduces the impact.

4   PROPOSED MODEL

The PRUS framework (Product Recommendation based on User Specification) follows a systematic process to extract aspect-level sentiments and generate a ranked product recommendation list.

- **Sentence Segmentation**

The process of sentiment analysis begins by breaking down customer reviews into individual sentences. This step is essential because reviews often express mixed opinions about different aspects of a product or service. For example, a customer might praise the design of a smartphone but complain about its battery life. By separating the review into distinct sentences, the system can evaluate each aspect independently, providing a more accurate assessment of the product's strengths and weaknesses. This granular analysis helps businesses better understand specific areas of satisfaction or dissatisfaction, rather than relying on an overall sentiment score, which can sometimes be misleading.

To achieve this sentence segmentation, the Python Natural Language Toolkit (NLTK) library is widely used. NLTK offers a variety of text processing tools, including tokenization, which is the process of dividing text into meaningful units. The sent_tokenize() function in NLTK is particularly useful for splitting text into sentences based on standard punctuation and linguistic patterns. This automated segmentation ensures that each sentence is treated as a separate unit, making it possible to analyze sentiments at a more detailed level. By isolating individual sentences, the system can avoid conflating conflicting opinions, leading to more accurate sentiment classification.

Sentence-level analysis is especially valuable because it addresses the complexity of human expression. Reviews often contain contradictory sentiments that, if analyzed as a whole, could result in an inaccurate or diluted sentiment score. For instance, a review stating, *"The laptop is incredibly fast, but the battery drains quickly,"* contains both positive and negative sentiments. When processed sentence by sentence, the system can correctly identify and categorize each opinion separately, offering a clearer picture of the product's performance.

Furthermore, this approach enhances feature-specific sentiment detection. For businesses, understanding which features customers appreciate or dislike is crucial for making informed decisions. By evaluating sentiments sentence by sentence, companies can identify trends, such as consistent complaints about delivery times or frequent praise for customer support. This detailed feedback allows for targeted improvements and more effective marketing strategies.

In summary, breaking reviews into individual sentences using NLTK significantly enhances the accuracy and effectiveness of sentiment analysis. By isolating specific opinions, the system can deliver more precise insights into customer satisfaction, helping businesses make data-driven improvements to their products and services.

- **Sentiment Analysis and Feature Extraction**

    In the second phase of the process, sentiment analysis and feature extraction are applied to the previously segmented sentences. This step is essential for evaluating the emotional tone of each sentence while simultaneously identifying the specific features being discussed. By analyzing sentences individually, the system can detect both positive and negative opinions on different aspects of a product or service. This granular approach offers more precise insights compared to analyzing the review as a whole, as it captures the sentiment of each distinct feature separately.

    To carry out this task, the Python **TextBlob** library is used. TextBlob is a popular natural language processing (NLP) tool known for its simplicity and effectiveness in performing sentiment classification and feature extraction. Its built-in functions allow for easy integration into larger sentiment analysis pipelines, making it a convenient choice for this phase. By leveraging TextBlob, the system can efficiently classify the sentiment polarity of each sentence while extracting relevant features mentioned by the user.

    When a sentence is processed, TextBlob assigns a **polarity score** that indicates the overall sentiment of the sentence. The polarity ranges from **-1.0** to **1.0**, where negative values represent negative sentiment, positive values denote positive sentiment, and scores around **0** indicate a neutral stance. For example, the sentence *"The app is easy to use but crashes frequently"* would receive both positive and negative sentiment scores for different aspects, allowing for a more balanced evaluation of the product.

In addition to sentiment classification, TextBlob performs **subjectivity analysis**, which measures how opinion-based or factual a sentence is. The subjectivity score ranges from **0.0** (completely objective) to **1.0** (entirely subjective). This helps differentiate between sentences that express clear opinions, such as *"I love the app's design"*, versus factual statements like *"The app was last updated in January."* This distinction is important when interpreting reviews, as subjective statements often reflect personal preferences, while objective ones describe verifiable facts.

Simultaneously, TextBlob handles **feature extraction** by identifying **noun phrases** in each sentence. Noun phrases represent the specific features being discussed, such as "battery life," "video quality," or "user interface." This allows the system to link the detected sentiment to the corresponding feature. For instance, if a sentence states, *"The battery drains too quickly,"* TextBlob would extract "battery" as the feature and classify the sentence as negative, providing detailed feedback on that particular aspect.

Overall, this phase plays a vital role in refining the sentiment analysis process. By applying TextBlob to both classify the sentiment and extract the mentioned features, the system gains a deeper understanding of user opinions. This detailed analysis makes it possible to offer more meaningful insights, such as identifying which product features receive consistent praise or complaints, ultimately enhancing the accuracy and value of the sentiment analysis process.

For example:
- In the sentence, *"The phone has an excellent camera but poor battery life"*, the features "camera" and "battery life" are extracted.
- The sentiment polarity is positive for the camera and negative for the battery life.
    - This process is applied to all the product reviews in the dataset. Each sentence is analyzed to extract both the features and their associated sentiments, which are appended together.
- Labelling Features with Sentiment
    - Once the features and their sentiments are extracted, they are labelled accordingly:
- If a sentence expresses positive sentiment, the associated features are marked as positively reviewed.

- Sentences classified as neutral result in the features being labelled as neutrally reviewed.
    - This labelling enables the system to quantify the sentiment distribution for each product feature, providing a detailed sentiment breakdown.
- Sentiment Scoring
    - The system then calculates sentiment scores by counting the number of times each feature is labelled as positive, negative, or neutral across all reviews for a specific product.
- The total positive, negative, and neutral counts for each feature are aggregated.
- The sentiment score for a feature is computed using the following formula:
    - *Score = (Positive count−Negative count) / Total Count*
- This formula ensures that features with more positive occurrences receive higher scores, while those with frequent negative mentions receive lower scores.
- The neutral mentions balance the score by preventing extreme polarity shifts.
    - For instance, if the feature "camera" appears 100 times in the reviews, with 60 positive mentions, 30 negative mentions, and 10 neutral mentions, the score would be:

      Score = (60-30) / 100 = 0.3
    - This indicates a moderately positive sentiment for the camera.

- **Ranking with RANK-ify Algorithm**

    - Finally, the system applies the RANK-ify algorithm to generate the ranked product recommendation list. The algorithm uses the calculated sentiment scores to rank the products:
    - Products with higher positive sentiment scores for the user-specified features are ranked higher. Products with more negative mentions or lower sentiment scores are ranked lower.
    - The RANK-ify algorithm ensures that the recommendations reflect both user preferences and sentiment-based feature evaluations, providing a personalized and accurate ranking.
    - Granular Sentiment Analysis: By analyzing sentence-level sentiments, the system captures aspect-specific opinions, preventing misleading overall polarity judgments.

o   Comprehensive Feature Scoring: The system quantifies the positivity and negativity of each feature, making the ranking more accurate and meaningful.

o   Balanced Recommendations: By considering both positive and negative sentiments, the framework provides realistic and reliable product recommendations.

4.1 Rank – ify Algorithm

The RANK-ify algorithm is a fundamental component of the PRUS (Personalized Relevance-based User Scoring) framework, designed to enhance product ranking by considering feature-specific scores based on user preferences. Unlike traditional ranking methods that rely on overall product ratings, RANK-ify prioritizes products that exhibit strong positive sentiment for the specific features a user values. This approach ensures that the ranking is highly personalized, helping users find products that align with their individual needs and preferences rather than simply highlighting the most popular or highest-rated items.

At its core, the RANK-ify algorithm processes the sentiment scores associated with product features, evaluating both positive and negative occurrences. When a user submits a query specifying the features they care about—such as "battery life" or "camera quality" in the case of smartphones—the algorithm analyzes the sentiment associated with these features across multiple product reviews. Products with consistently positive feedback for the specified features are ranked higher, while those with frequent negative mentions are pushed lower in the list. This dual consideration of both positive and negative sentiment ensures a balanced and realistic ranking, providing users with a clearer picture of product performance.

One of the key strengths of RANK-ify lies in its adaptability. The algorithm dynamically adjusts the ranking based on the user's preferences, giving greater weight to features they prioritize. For example, if a user values "battery life" over "camera quality" when searching for a smartphone, RANK-ify will prioritize products with stronger positive sentiment regarding battery performance, even if they have slightly lower camera scores. This personalized weighting makes the recommendations more relevant to individual users, enhancing their overall shopping experience.

Additionally, RANK-ify considers the frequency and intensity of feature mentions. Products with numerous, consistently positive mentions for a given feature are ranked higher than those with sporadic or lukewarm feedback. This frequency-based weighting ensures that products with consistently high-performing features receive better visibility. On the other hand, products with mixed or inconsistent feedback are ranked lower, preventing misleading or overly optimistic recommendations.

The RANK-ify algorithm also addresses the challenge of conflicting sentiments by applying a sentiment balancing mechanism. In cases where products receive both positive and negative mentions for the same feature, the algorithm evaluates the overall sentiment trend. For example, if a laptop receives ten positive mentions about its speed and two negative mentions, the algorithm balances the sentiment by assigning it an appropriate score that reflects the overall customer experience, rather than being skewed by isolated negative feedback.

In summary, the RANK-ify algorithm is a powerful tool for generating relevance-based product rankings tailored to individual user preferences. By analyzing both positive and negative occurrences of features, applying dynamic weighting, and considering sentiment consistency, RANK-ify offers a realistic and user-centric product ranking system. This ensures that users receive more accurate and meaningful recommendations, helping them make informed purchasing decisions.

4.1.1 Input Parameters and Initialization
The RANK-ify algorithm takes three input parameters:
- **k**: Represents the maximum number of products to be returned in descending order of relevance, i.e., from the most relevant to the least relevant according to the user's query.
- **Lp**: Refers to the set of all products in the dataset that have been reviewed. This set contains the features associated with each product along with the positive and negative occurrence counts of these features.
- **Q:** Represents the list of features extracted from the user's query, specifying the aspects that the user prioritizes in the product search.

4.1.2 Iterative Feature Matching
- Every feature of each product (Line 4) and
- Each feature in the user query (Line 5).

For each product, the algorithm checks if the user-specified feature exists in the list of features associated with the product (Line 6). If the queried feature is present, the algorithm proceeds to calculate the Feature Score (FeaSco) for that feature using Equation 1 (Line 7):

$$FeaSco(f)=(w1*c(f+)-w2*c(f-))\,/\,(c(f+)+c(f-))$$

Where:

- $c(f+) \rightarrow$ The **count of positive occurrences** of the feature fff.
- $c(f-) \rightarrow$ The **count of negative occurrences** of the feature.
- $w1 \rightarrow$ The **weight assigned to positive occurrences**.
- $w2 \rightarrow$ The **weight assigned to negative occurrences**.

4.1.3 Weighted Sentiment Scoring

The weight attributes w1 and w2 control the impact of positive and negative occurrences on the final ranking. Since users may value positive and negative opinions differently, these weights allow the system to adjust the influence of each sentiment.

- For example, if the user is more concerned with positive feedback, w1 will be larger than w2.
- Conversely, if the user gives more importance to negative feedback, w2 will be larger.
- To maintain balance and prevent one sentiment from dominating the ranking excessively, the algorithm enforces the condition:

$$w1+w2=1$$

This ensures that the influence of both sentiment polarities is normalized.

4.1.4 Rank-Score Calculation

Once the FeaSco is calculated for each user-specified feature in every product, the algorithm sums these feature scores to compute the overall Rank-Score of the product (Line 10). This Rank-Score represents the combined sentiment impact of all features in relation to the user's query.

- Products with higher Rank-Scores exhibit stronger positive alignment with the user's preferences.
- Products with more negative sentiment or fewer matching features receive lower scores.

Sorting and Final Recommendation

After calculating the Rank-Score for all products, the algorithm sorts the products in descending order (Line 12), prioritizing those with higher Rank-Scores.

- The top-k products from the sorted list are returned to the user (Line 13), providing a personalized and ranked recommendation list that closely matches their preferences.

Example Execution

For example, if a user queries for a smartphone with good battery life and camera, the algorithm:

- Iterates through all products and checks if they contain these features.
- Calculates the FeaSco for both battery life and camera by considering the positive and negative occurrences.
- Combines the FeaSco values into an overall Rank-Score for each product.
- Sorts the products based on their Rank-Score and presents the top-k ranked products to the user.

Algorithm 1 RANK-Ify Algorithm

1: k =Number of Products in Dataset
2: L_p =List of Products in Dataset
3: Q =Features in User Query
4: **for** $p \in L_p$ **do**
5:   **for** $U_f \in Q$ **do**
6:     **if** $U_f \in p$ **then**
7:       $FS(U_f) = FeaSco(U_f)$
8:     **end if**
9:   **end for**
10:   $RS(p) = \sum_{U_f \in Q} FS(U_f)$
11: **end for**
12: Sort L_p w.r.t. RS
13: Return top-k from L_p

Fig. 4.1.4.1

4.2 Key Benefits of the RANK-ify Algorithm

- **Granular Sentiment Evaluation:** By considering both positive and negative occurrences, the algorithm offers a balanced ranking.

- **Weighted Scoring:** The weight parameters allow the system to customize the impact of positive and negative feedback, aligning with user priorities.

- **Efficient and Scalable:** The algorithm efficiently processes large datasets by iterating over relevant features only, making it suitable for scalable e-commerce platforms.

- **Personalized Recommendations:** By prioritizing products based on user-specified features, the algorithm generates accurate and personalized recommendations, enhancing the user experience.

Overall, the **RANK-ify algorithm** is a robust and efficient method for ranking products based on aspect-level sentiment analysis.

5 DISCUSSIONS ON THE PROPOSED OUTPUT

5.4 Experiments and Results

The experimental evaluation of the PRUS framework demonstrates its effectiveness in generating personalized product recommendations by ranking products based on aspect-level sentiment analysis. The evaluation was conducted using a preprocessed Amazon dataset containing mobile phone reviews from various brands. The system was tested by extracting features from a user query and generating a ranked recommendation list based on the Discounted Cumulative Gain (DCG) measure. The results highlight the accuracy and reliability of the PRUS framework in providing feature-specific recommendations, overcoming the limitations of overall review-based scoring.

Sentiment-Driven Feature Scoring

In the proposed approach, TextBlob was used to perform sentiment analysis, assigning individual sentiment scores to the features of each product. These scores were normalized on a scale of 1 to 10 to ensure consistent evaluation across all products. The framework calculates the Average Aspect Score (AAS) for each product usingthebelow equation:

$$AAS(p, Q) = 1/Count(\sum_{i=1}^{count} S(i))$$

Here, for a product p, all the features relevant to the user query (Q) are considered. Their sentiment scores are averaged, providing a composite score that reflects the product's overall feature relevance. This ensures that products are ranked based on the sentiment scores of the features that matter most to the user, rather than relying on an overall rating.

For example, in a review stating:

"The screen quality is good, but the battery life is poor,"

The **screen quality** receives a **sentiment score of 8.65**, while the **battery life is assigned score of 3.7**, indicating positive and negative sentiments for different features. Traditional systems would assign a single overall rating, which could be misleading, while PRUS assigns individual scores to each feature, offering more precise and granular sentiment analysis.

DCG and nDCG-Based Ranking

Once the AAS values are calculated for all products, the DCG score is computed using the below Equation:

$$DCG_p = AAS_1(p, Q) + \sum_{a=1}^{p} \frac{AAS_a(p, Q)}{log_2 a}$$

The DCG score quantifies how effectively a product matches the user-specified features and their associated sentiment scores. To ensure a fair and unbiased ranking, the system calculates the Ideal DCG (IDCG) using the below Equation:

$$IDCG_p = \sum_{i=1}^{p} \frac{AAS_i}{log_2(i + 1)}$$

The IDCG represents the optimal ranking scenario, where the most relevant products receive the highest scores. Finally, nDCG is calculated using the below Equation:

$$nDCG_P = \frac{DCG_P}{IDCP_P}$$

The nDCG value ensures that the ranking is normalized, making it more consistent and reliable across different product lists.

Demonstration of the Output

To demonstrate the effectiveness of PRUS, the authors conducted an example evaluation using a user query that requested a mobile phone with a good camera and high screen resolution. The system successfully displayed the top 10 ranked products based on:

- Rank Score (RS): Calculated using the RANK-ify algorithm.
- nDCG values: Derived using Equation 5.

  The results showed that the highest-ranked products were those that received strong positive sentiment scores for both camera quality and screen resolution. While several products shared the same Rank Score (0.9), the nDCG values varied, allowing the system to further differentiate the products based on their overall sentiment relevance.

- The products with higher nDCG scores appeared at the top of the recommendation list, as they exhibited stronger positive feedback for the features specified by the user.
- The products with lower nDCG scores were ranked lower, reflecting their weaker alignment with the user's query.

Key Advantages of the PRUS Output

The experimental results demonstrate several key benefits of the PRUS framework:

- Feature-Specific Ranking: Unlike traditional models that rely on overall ratings, PRUS assigns individual sentiment scores to product features, ensuring more accurate and personalized recommendations.

- Granular Sentiment Analysis: The system successfully captures both positive and negative opinions about different product features, preventing misleading overall ratings.

- DCG-Based Normalization: The use of nDCG values ensures that products are ranked fairly and consistently, even when their Rank Scores are identical.

- Improved Decision-Making: The feature-based scoring allows users to make better-informed decisions by clearly identifying products that excel in the specific features they care about.

- Efficient Query-Based Personalization: PRUS effectively tailors the recommendations to user preferences, offering a highly customized product ranking.

Discussion on Practical Implications

The PRUS framework offers a significant improvement over traditional product ranking methods, especially for e-commerce platforms. The ability to prioritize features based on user-specified preferences ensures that the recommendations are:

- Highly relevant and tailored to individual needs.

- Accurate, as the system captures fine-grained feature sentiments rather than relying on general product ratings.

- Efficient, by applying DCG and nDCG scoring, making the ranking fair and consistent across products.

 For example, a customer searching for a laptop with long battery life and lightweight design would benefit from PRUS, as it ranks products based on aspect-specific sentiment scores rather than general ratings.

Overall, the experimental output of the PRUS framework demonstrates its effectiveness, reliability, and practicality in delivering accurate, feature-specific, and personalized product recommendations.

5.5 Performance Analysis of the PRUS Method

The PRUS framework underwent a series of experiments to rigorously evaluate its performance and effectiveness in generating ranked product recommendations based on user-specified features. These experiments were designed to test how accurately and efficiently the framework could identify and rank products according to individual feature preferences, ensuring that the recommendations aligned with specific user needs. The evaluation aimed to validate the framework's capability of providing precise and personalized suggestions, thereby enhancing the overall recommendation accuracy.

During the experiments, the weights w_1 and w_2 were systematically varied. These weights play a crucial role in the ranking process, as they control the influence of positive and negative occurrences of product features, respectively. The w_1 weight determines the impact of positive feature mentions, while the w_2 weight governs the effect of negative feature occurrences. By altering the values of w_1 and w_2, the framework's sensitivity to favorable and unfavorable product attributes could be finely tuned. This allowed the researchers to examine how the PRUS framework's ranking accuracy changed in response to different weight configurations.

To measure the framework's performance, two evaluation metrics were employed: the Rank Score (RS) and the Normalized Discounted Cumulative Gain (nDCG). The Rank Score measures the relevance and accuracy of the product rankings by comparing the framework-generated order to an ideal ranking based on ground-truth data. A higher RS indicates a closer alignment between the predicted and actual product preferences. The nDCG metric, on the other hand, evaluates the quality of the ranking by considering both the position of relevant items and their relevance scores. nDCG rewards models that place highly relevant products near the top of the list, making it a valuable indicator of ranking effectiveness.

Throughout the experimentation process, the impact of weight variations on both RS and nDCG was carefully analyzed. By systematically adjusting the positive and negative influence weights, the researchers observed how the ranking accuracy shifted. For instance, when w_1 was increased, products with more positive feature occurrences were ranked higher, whereas increasing w_2 gave more emphasis to products with fewer negative mentions. This allowed the team to assess whether giving greater weight to positive or negative features resulted in more effective product recommendations.

The experimental results demonstrated that the PRUS framework effectively adapts to different weight configurations, showcasing its flexibility and robustness. When properly tuned, the framework achieved high RS and nDCG scores, indicating that it accurately ranked products according to user-specified feature preferences. These findings confirmed that the framework could effectively prioritize products with desirable attributes while downranking those with less favorable characteristics, leading to more accurate and user-centric recommendations.

In summary, the experiments validated the PRUS framework's effectiveness in generating personalized product recommendations by dynamically adjusting the influence of positive and negative feature occurrences. The use of Rank Score and nDCG metrics provided concrete evidence of the framework's accuracy and efficiency. The ability to fine-tune the w_1 and w_2 weights ensures that the framework can be tailored to different user preferences, making it a valuable tool for e-commerce platforms seeking to provide precise and feature-specific product recommendations.

5.2.1 Experiment Setup and Weight Variations

In the experiments, the weights were alternated in such a way that their sum equals to one (1) , i.e.,w1+w2=1

The weights were varied over a range of 0.1 to 0.9, creating different combinations of positive and negative influence on the final product ranking. The product list (P1 to P10) represents the top-ranked products based on the RS and nDCG scores corresponding to the different weight combinations. The results of these experiments are summarized in the below Table.

Product		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Rank		1	2	3	4	5	6	7	8	9	10
PW=0.9	Rank Score	2.7	2.7	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8
NW=0.1	nDCG Score	0.94	2.06	2.81	3.59	4.06	4.44	4.94	4.94	5.81	6.67
PW=0.8	Rank Score	2.24	2.24	1.61	1.61	1.61	1.61	1.61	1.61	1.61	1.61
NW=0.2	nDCG Score	1.00	2.06	2.92	3.65	4.15	4.55	4.90	5.37	5.91	6.36
PW=0.7	Rank Score	2.1	2.1	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
NW=0.7	nDCG Score	1.00	2.06	3.06	3.59	3.77	4.57	4.85	5.27	5.50	6.44
PW=0.6	Rank Score	1.8	1.8	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
NW=0.4	nDCG Score	0.94	2.06	2.79	3.60	4.02	4.44	4.58	5.15	5.97	6.72
PW=0.5	Rank Score	1.5	1.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
NW=0.5	nDCG Score	0.94	2.06	3.07	3.39	4.02	4.57	4.84	5.47	5.70	6.03
PW=0.4	Rank Score	1.2	1.2	1	1	1	0.85	0.84	0.82	0.80	0.80
NW=0.6	nDCG Score	1	2.06	2.73	3.29	3.47	4.57	5.41	6.13	7.36	8.07
PW=0.3	Rank Score	1	1	1	0.9	0.9	0.71	0.7	0.7	0.68	0.65
NW=0.7	nDCG Score	0.488	1.04	1.97	4.05	5.40	4.62	2.23	2.65	14.9	15.8
PW=0.2	Rank Score	1	1	1	0.8	0.8	0.6	0.6	0.6	0.57	0.52
NW=0.8	nDCG Score	0.48	1.00	2.09	1.42	1.99	6.67	8.35	6.00	14.7	17.2
PW=0.1	Rank Score	1	1	0.9	0.9	0.63	0.5	0.42	0.360	0.36	0.52
NW=0.9	nDCG Score	1.89	2.65	1.93	3.60	3.97	5.64	13.2	8.01	9.13	17.2

Table No. 5.2.1.1

5.2.2 Impact of Positive and Negative Weights on nDCG

The nDCG scores represent the relevance of the ranked products based on the user-specified features and their associated sentiments. The experimental results reveal the following trends:

- When the positive weight (w1) ranges between 0.9 and 0.5, there is no significant change in the overall nDCG scores. This indicates that as long as positive sentiments dominate, the relevance of the product ranking remains consistent.

- However, as soon as the negative weight (w2) increases beyond 0.5, there is a sudden increase in the nDCG scores. This happens because products with more negative mentions gain higher weight, making them more prominent in the ranking.

- This outcome demonstrates that nDCG scores increase when negative feedback is given more importance, reflecting a shift in product relevance towards products with more critical sentiments.

5.2.3 Impact on Rank Score (RS)

The **Rank Score (RS)**, calculated using the **RANK-ify algorithm**, represents the overall product ranking based on both positive and negative occurrences of the features. The experimental results show the following trend:

- When the positive weight (w1) is high (greater than 0.5), products with more positive feature sentiments receive higher Rank Scores.

- As the negative weight (w2) crosses 0.5, the RS begins to decline. This is because the increased influence of negative feedback causes products with more negative mentions to be ranked higher, lowering the RS for products with predominantly positive reviews.

- This behaviour aligns with the FeaSco formula (Equation 1), which defines the Feature Score as:

$$FeaSco(f) = (w1 * c(f+) - w2 * c(f-)) / (c(f+) + c(f-))$$

Where:

- w1 → The positive weight.
- w2 → The negative weight.
- c(f+) → The count of positive occurrences.
- c(f−) → The count of negative occurrences.
- Increasing w2 reduces the overall FeaSco value, leading to lower Rank Scores for products with more positive mentions.

5.2.4 Influence of Weight Variation on the Product List

The product list (P1 to P10) varies significantly as the positive and negative weights are adjusted.

- With higher positive weights, the products with strong positive sentiment for the user-specified features dominate the top ranks.

- As the negative weight increases, the list becomes populated with products that have more negative mentions for the specified features, reflecting the increased influence of negative sentiment.

- This highlights the flexibility of the PRUS framework, allowing users to adjust the ranking by controlling the impact of positive and negative occurrences.
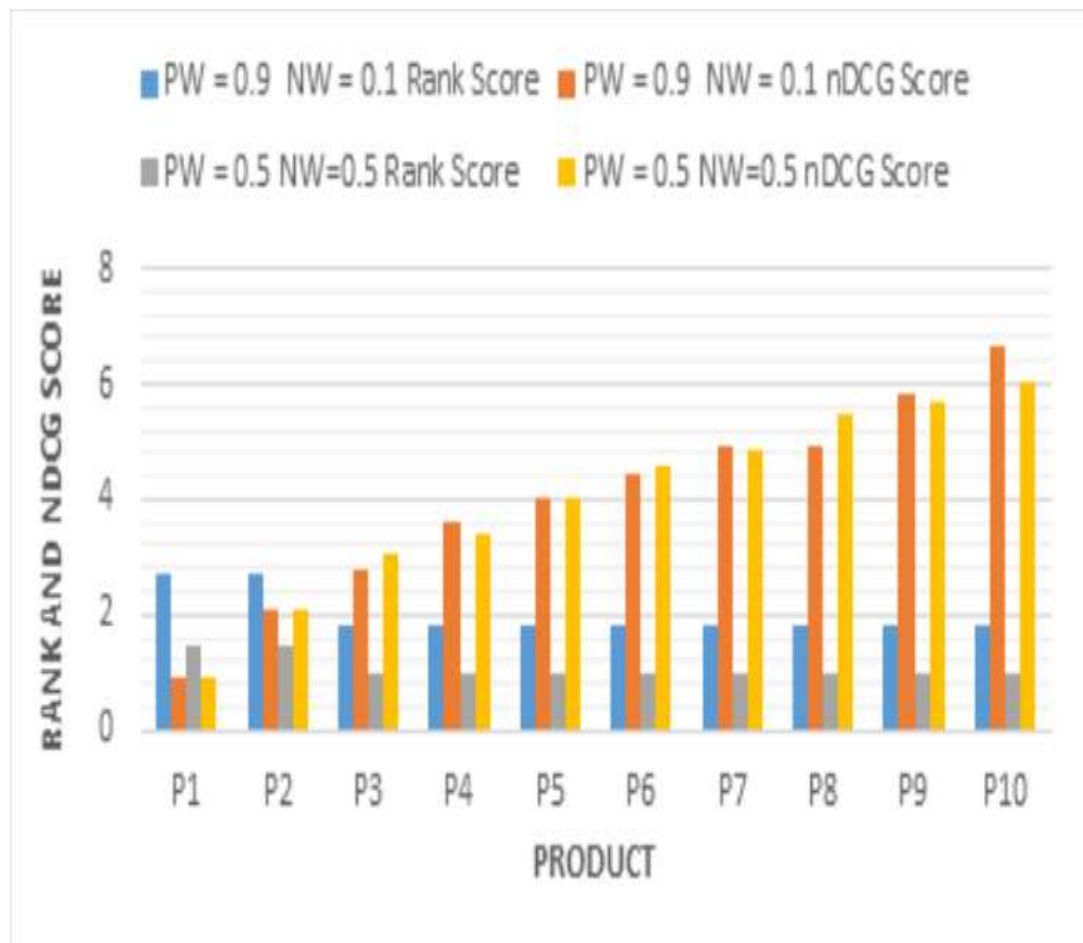

Fig. 5.2.4.1

5.2.5 Key Observations from the Performance Analysis

The experiments reveal several key insights into the performance of the PRUS framework that was discussed in the paper:

- No Major Change in nDCG with Higher Positive Weight: When the positive weight ranges between 0.9 and 0.5, there is no substantial change in the nDCG scores. This suggests that as long as positive sentiment dominates, the overall relevance of the recommendations remains stable.

- nDCG Increases with Higher Negative Weight: As the negative weight increases beyond 0.5, nDCG scores rise sharply. This is because the system prioritizes products with more negative feedback, making critical reviews more prominent in the ranking.

- Rank Scores Decrease with Increasing Negative Weight: When negative sentiment gains more influence ($w2 > 0.5$), the Rank Scores decline. This occurs because negative features are weighted more heavily, reducing the overall RS of products with predominantly positive reviews.

- User-Controlled Sentiment Influence: The PRUS framework provides users with the ability to control the degree of positive and negative influence on the final product list. This makes the system highly flexible and customizable, catering to individual preferences.

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this study, a Product Recommendation based on User Specification (PRUS) framework was proposed to generate a personalized, ranked list of products by analyzing customer reviews. Unlike existing methods, the PRUS approach incorporates both positive and negative sentiments of individual product features, enabling more accurate and personalized recommendations. While traditional ranking models often prioritize products based on overall sentiment, PRUS addresses the granularity issue by performing aspect-level sentiment analysis, ensuring that feature-specific opinions influence the ranking process.

The PRUS framework operates in three phases:

1. **Feature and Sentiment Extraction**: The system extracts features and their associated sentiments from individual sentences of customer reviews, capturing both positive and negative opinions at the sentence level.

2. **Feature-Sentiment Association**: The extracted features are appended with their respective sentiment polarities, allowing the system to quantify the positive and negative occurrences of each feature.

3. **Ranking Phase**: The user-specified features are extracted from the query, and the products are ranked using the RANK-ify algorithm based on their Rank Scores (RS). The RS considers both positive and negative weights, providing a balanced and realistic ranking.

   The PRUS framework offers flexibility by allowing users to adjust the weight parameters associated with positive and negative sentiments. This gives users the ability to customize the ranking based on their individual preferences. Through extensive experiments, the PRUS framework demonstrated its effectiveness and accuracy in generating personalized recommendations, showing that:

- Increasing the negative weight results in a higher nDCG score, as products with more negative mentions become more prominent in the ranking.

- Increasing the positive weight causes the Rank Score to increase, prioritizing products with more positive sentiment.

  The experimental results validated the reliability and accuracy of the PRUS framework, making it a robust solution for e-commerce platforms and other review-based product ranking systems.

6.2 Future Work

While the PRUS framework has demonstrated promising results, several avenues for future paper remain open. These directions aim to enhance the framework's flexibility, scalability, and applicability in real-world scenarios:

1. **Multilingual Product Review Datasets**:
   o The current PRUS framework is applied to English-language reviews only. A natural extension of this work would be to implement multilingual support.
   o This would enable the framework to process reviews in multiple languages, making it applicable to global e-commerce platforms.
   o Future work could incorporate language translation models or multilingual sentiment analysis techniques to extract feature-level sentiments across diverse languages.

2. **User-Controlled Polarity Customization**:
   o The PRUS framework currently allows users to adjust the weight parameters associated with positive and negative sentiments, offering some flexibility in ranking.
   o Future enhancements could provide more advanced customization options, allowing users to:
      ▪ Specify individual sentiment thresholds for specific features.
      ▪ Define custom polarity preferences, such as favoring features with balanced sentiments or discounting neutral feedback.
   o This would result in more dynamic and user-centric recommendations.

3. **Incorporating Contextual and Temporal Factors**:
   o Future paper could enhance the PRUS framework by incorporating contextual information such as time-sensitive reviews or trending sentiments.
   o This would allow the system to prioritize recent reviews or consider seasonal trends when ranking products, making the recommendations more relevant and timely.

4. **Real-Time Product Recommendation System**:
   o To make the PRUS framework more practical for real-world applications, future work could focus on real-time product ranking.
   o This would enable the system to dynamically rank products as new reviews are added, ensuring that the recommendations remain current and accurate.

REFERENCES

[1] P. K. Jain, R. Pamula, and G. Srivastava, ''A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews,'' Comput. Sci. Rev., vol. 41, Aug. 2021, Art. no. 100413.

[2] S. Moro and P. Rita, ''Data and text mining from online reviews: An automatic literature analysis,'' WIREs Data Mining Knowl. Discovery, vol. 12, no. 3, May 2022, Art. no. e1448. [3] N. Hussain, H. T. Mirza, G. Rasool, I. Hussain, and M. Kaleem, ''Spam review detection techniques: A systematic literature review,'' Appl. Sci., vol. 9, no. 5, p. 987, Mar. 2019.

[4] L. He, X. Wang, H. Chen, and G. Xu, ''Online spam review detection: A survey of literature,'' Hum.-Centric Intell. Syst., vol. 2, nos. 1–2, pp. 14–30, Jun. 2022.

[5] S. K. Maurya, D. Singh, and A. K. Maurya, ''Deceptive opinion spam detection approaches: A literature survey,'' Int. J. Speech Technol., vol. 53, no. 2, pp. 2189–2234, Jan. 2023.

[6] E. Lee, F. Rustam, P. B. Washington, F. E. Barakaz, W. Aljedaani, and I. Ashraf, ''Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble GCR-NN model,'' IEEE Access, vol. 10, pp. 9717–9728, 2022.

[7] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, ''Automatic text summarization: A comprehensive survey,'' Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 113679.

[8] M. Umer, I. Ashraf, A. Mehmood, S. Kumari, S. Ullah, and G. S. Choi, ''Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model,'' Comput. Intell., vol. 37, no. 1, pp. 409–434, Feb. 2021.

[9] P. K. Soni and R. Rambola, ''A survey on implicit aspect detection for sentiment analysis: Terminology, issues, and scope,'' IEEE Access, vol. 10, pp. 63932–63957, 2022.

[10] N. Aslam, K. Xia, F. Rustam, A. Hameed, and I. Ashraf, ''Using aspectlevel sentiments for calling app recommendation with hybrid deep-learning models,'' Appl. Sci., vol. 12, no. 17, p. 8522, Aug. 2022.

[11] G. Brauwers and F. Frasincar, ''A survey on aspect-based sentiment classification,'' ACM Comput. Surv., vol. 55, no. 4, pp. 1–37, Apr. 2023.

[12] M. Sivakumar and S. R. Uyyala, ''Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic,'' Int. J. Data Sci. Anal., vol. 12, no. 4, pp. 355–367, Oct. 2021.

[13] J. Zhang, X. Lu, and D. Liu, ''Deriving customer preferences for hotels based on aspect-level sentiment analysis of online reviews,'' Electron. Commerce Res. Appl., vol. 49, Sep. 2021, Art. no. 101094.

[14] K. Ganesan and C. Zhai, ''Opinion-based entity ranking,'' Inf. Retr., vol. 15, no. 2, pp. 116–150, Apr. 2012.

[15] L. Akoglu, ''Opinion fraud detection in online reviews by network effects,'' in Proc. ICWSM, vol. 7, no. 1, 2013, pp. 2–11.

[16] E. Marrese-Taylor, J. D. Velásquez, and F. Bravo-Marquez, ''A novel deterministic approach for aspect-based opinion mining in tourism products reviews,'' Expert Syst. Appl., vol. 41, no. 17, pp. 7764–7775, Dec. 2014.

[17] A. Kumar and S. Abirami, ''Aspect-based opinion ranking framework for product reviews using a Spearman's rank correlation coefficient method,'' Inf. Sci., vols. 460–461, pp. 23–41, Sep. 2018.

[18] S. Bashir, W. Afzal, and A. R. Baig, ''Opinion-based entity ranking using learning to rank,'' Appl. Soft Comput., vol. 38, pp. 151–163, Jan. 2016.

[19] T. A. Rana and Y.-N. Cheah, ''A two-fold rule-based model for aspect extraction,'' Expert Syst. Appl., vol. 89, pp. 273–285, Dec. 2017.

[20] C. Wu, F. Wu, S. Wu, Z. Yuan, and Y. Huang, ''A hybrid unsupervised method for aspect term and opinion target extraction,'' Knowl.-Based Syst., vol. 148, pp. 66–73, May 2018.

[21] Z. Luo, S. Huang, F. F. Xu, B. Y. Lin, H. Shi, and K. Zhu, ''ExtRA: Extracting prominent review aspects from customer feedback,'' in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 3477–3486.

[22] S. Sharmila and S. Vijayarani, ''Association rule mining using fuzzy logic and whale optimization algorithm,'' Soft Comput., vol. 25, no. 2, pp. 1431–1446, Jan. 2021.

[23] S. Gurumoorthy, B. N. K. Rao, X.-Z. Gao, B. V. Krishna, A. K. Pandey, and A. S. Kumar, ''Feature based opinion mining and sentiment analysis using fuzzy logic,'' in Cognitive Science and Artificial Intelligence (SpringerBriefs in Forensic and Medical Bioinformatics). Singapore, 2018, pp. 79–89.

[24] M. Venugopalan and D. Gupta, ''An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis,'' Knowl.-Based Syst., vol. 246, Jun. 2022, Art. no. 108668.

[25] M. Hu and B. Liu, ''Mining and summarizing customer reviews,'' in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2004, pp. 168–177.

[26] A.-M. Popescu and O. Etzioni, ''Extracting product features and opinions from reviews,'' in Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT), 2005, pp. 9–28.

[27] S. Li, L. Zhou, and Y. Li, ''Improving aspect extraction by augmenting a frequency-based method with web-based similarity measures,'' Inf. Process. Manage., vol. 51, no. 1, pp. 58–67, Jan. 2015.

[28] Z. Li, M. Zhang, S. Ma, B. Zhou, and Y. Sun, ''Automatic extraction for product feature words from comments on the web,'' in Proc. Asia Inf. Retr. Symp. Sapporo, Japan: Springer, Oct. 2009, pp. 112–123.

[29] S. Raju, P. Pingali, and V. Varma, ''An unsupervised approach to product attribute extraction,'' in Proc. Eur. Conf. Inf. Retr. Toulouse, France: Springer, Apr. 2009, pp. 796–800.

[30] Ö. F. Güneş, T. Furche, and G. Orsi, ''Structured aspect extraction,'' in Proc. COLING 2016, 26th Int. Conf. Comput. Linguistics, Tech. Papers, 2016, pp. 2321–2332.

[31] C. Quan and F. Ren, ''Unsupervised product feature extraction for feature-oriented opinion determination,'' Inf. Sci., vol. 272, pp. 16–28, Jul. 2014.

[32] W. Maharani, D. H. Widyantoro, and M. L. Khodra, ''Clue propagation based on non-adjective opinion words for handling disconnected propagation in product reviews,'' IEEE Access, vol. 10, pp. 56785–56795, 2022.

[33] X. Luo, ''Efficient english text classification using selected machine learning techniques,'' Alexandria Eng. J., vol. 60, no. 3, pp. 3401–3409, Jun. 2021.

[34] C.-P. Wei, Y.-M. Chen, C.-S. Yang, and C. C. Yang, ''Understanding what concerns consumers: A semantic approach to product feature extraction from consumer reviews,'' Inf. Syst. e-Bus. Manage., vol. 8, no. 2, pp. 149–167, Mar. 2010.

[35] B. Ma, D. Zhang, Z. Yan, and T. Kim, ''Anlda and synonym lexicon based approach to product feature extraction from online consumer product reviews,'' J. Electron. Commerce Res., vol. 14, no. 4, p. 304, 2013.

[36] K. Liu, L. Xu, and J. Zhao, ''Co-extracting opinion targets and opinion words from online reviews based on the word alignment model,'' IEEE Trans. Knowl. Data Eng., vol. 27, no. 3, pp. 636–650, Mar. 2015. 81296 VOLUME 11, 2023 N. Hussain et al.: PRUS: Product Recommender System Based on User Specifications and Customers Reviews.

[37] C.-H. Lai, D.-R. Liu, and K.-S. Lien, ''A hybrid of XGBoost and aspectbased review mining with attention neural network for user preference prediction,'' Int. J. Mach. Learn. Cybern., vol. 12, no. 5, pp. 1203–1217, May 2021.

[38] G. Kaur and A. Sharma, ''A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis,'' J. Big Data, vol. 10, no. 1, pp. 1–23, Jan. 2023. [39] M. Nasir and C. I. Ezeife, ''Semantic enhanced Markov model for sequential E-commerce product recommendation,'' Int. J. Data Sci. Anal., vol. 15, no. 1, pp. 67–91, Jan. 2023.

[40] S. Bird, E. Klein, and E. Loper, Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit. Sebastopol, CA, USA: O'Reilly Media, 2009.

[41] S. Loria, Textblob Documentation, document Release 0.15, 2018, vol. 2, no. 8.

[42] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells, ''Assessing ranking metrics in top-N recommendation,'' Inf. Retr. J., vol. 23, no. 4, pp. 411–448, Aug. 2020.

[43] P. Nitu, J. Coelho, and P. Madiraju, ''Improvising personalized travel recommendation system with recency effects,'' Big Data Mining Anal., vol. 4, no. 3, pp. 139–154, Sep. 2021. [44] M. K. Singh and O. P. Rishi, ''Event driven recommendation system for E-commerce using knowledge based collaborative filtering technique,'' Scalable Comput., Pract. Exper., vol. 21, no. 3, pp. 369–378, Aug. 2020.

[45] Z.-P. Fan, G.-M. Li, and Y. Liu, ''Processes and methods of information fusion for ranking products based on online reviews: An overview,'' Inf. Fusion, vol. 60, pp. 87–97, Aug. 2020.