Journal of Harbin Engineering University Vol 44 No. 7
ISSN: 1006-7043 July 2023

An Opinion Mining-Based Hybrid Collaborative Filtering Recommendation Model for Consumer Decision

Analp Pathak[1,2], Dr. B. K. Sharma[3]
[1]Research Scholar, Dr. A. P. J. Abdul Kalam technical University, Lucknow
[2]Department of Information Technology, KIET Group of Institutions, Delhi-NCR, Ghaziabad, India
[3]Principal Scientific Officer and Head Computer Science and Engineering Department, NITRA Technical Campus, Ghaziabad

Abstract:
A variety of information sources can be utilized by a recommendation system to suggest items that align with the diverse interests of its users. Typically, such systems employ the collaborative filtering (CF) method, which involves combining a user's preference data with that of others to anticipate additional items that may pique their interest. This research proposes a novel weighted recommendation system based on CF to facilitate better decision-making among consumers. The study presents two equations, one for calculating the weight of a product and its review, and another for determining the similarity between reviews provided by different consumers. The methodology utilizes a combination of Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR) ensemble classifiers to improve model performance. The proposed model is trained and evaluated using an open-source dataset available on Kaggle's website. The numerical analysis indicates that the proposed model outperforms other conventional methods in terms of accuracy (0.821), precision (0.802), recall (0.821), F-measure (0.833), and error rate (0.100), among other metrics.
Keywords: Recommendation System, CF, Multi-nomial Naïve Bayes, Multi-Layer Perceptron, Logistic Regression, and Ensemble Classifiers

1. Introduction

The age of big data has caused a rapid influx of information affecting online businesses. Amazon alone attracts an average of 900 million customers every day, leading to an information explosion for e-commerce users. To address this issue, recommendation systems are used to analyze data from users' previous interactions and determine their preferences and needs, assisting them in making informed decisions. In recent years, the majority of research efforts in this area have focused on recommendation systems [1][2][3].

One effective way to manage this overwhelming amount of information is to rely on recommendations from other individuals who encounter similar data challenges. Recommendations can be conveyed through various mediums such as spoken words, letters of recommendation, news reports, public surveys, travel guides, website evaluations, and more. For the past 15 years, several major electronic sites have implemented recommendation systems to facilitate this natural social process. These systems aim to assist consumers in finding the most relevant and useful information among the vast amount of material accessible online, including news articles, web pages, pictures, and more [4].

1.1 Recommendation System

A recommendation system is an electronic tool that assists customers in identifying the most suitable items or services based on their past preferences or tastes. These preferences or tastes can be deduced from customers' purchase histories. As online business continues to expand, recommender systems are becoming increasingly important for personalized marketing. A well-designed recommender system can analyze the preferences inferred or explicitly expressed by each customer and automatically provide a selection of items or services [5].

Using a parallel strategy to generate recommendations has several benefits. One such advantage is the ability to deliver results quickly. Parallel execution of algorithms can generate output more efficiently without sacrificing performance. Other benefits include the ability to handle large volumes of data quickly and suggest a wide range of item types. Additionally, it is

straightforward to convert an existing method to parallel processing [6]. Figure 1 shows the three stages of a recommended system:
Data collection: The system collects data on user preferences and behaviour.
Filtering: The collected data is processed using algorithms to filter out irrelevant information and identify patterns in user preferences.
Recommendation: Based on the filtered data, the system generates recommendations for the user.
Overall, a parallel recommendation system is an efficient and effective way to generate personalized recommendations for customers.

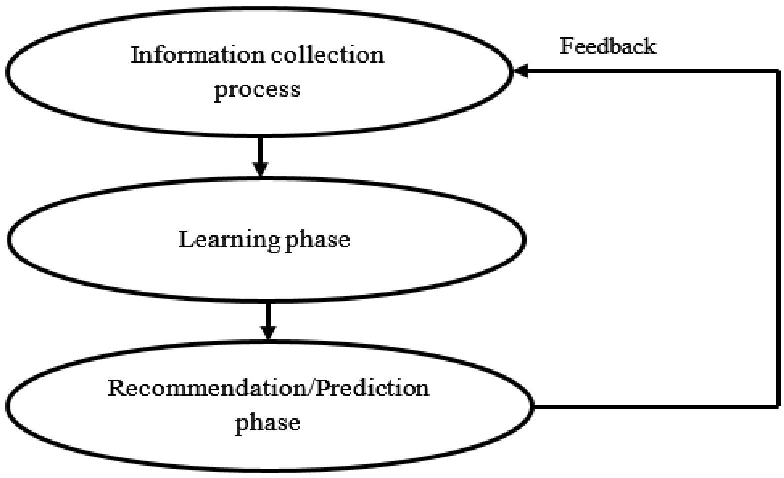

Figure 1. Recommendation System.

When it comes to making recommendations, there are two main approaches: content-based and collaborative filtering (CF).
The content-based approach involves identifying users by attempting to identify their key characteristics, which requires individual information that is difficult to obtain. This approach involves analyzing the characteristics of items that users have liked in the past, and recommending other items with similar characteristics.
On the other hand, the CF approach takes advantage of the fact that people who have shared interests in the past are likely to have similar tastes in the future. CF involves analyzing the preferences of other users with similar interests, and recommending items that those users have liked in the past.
Both approaches have their strengths and weaknesses, and the choice between them depends on the specific application and the available data. Content-based approaches can be effective when there is a limited amount of user data available, but they may not capture the full range of a user's interests. CF approaches, on the other hand, can be effective when there is a large amount of user data available, but they may suffer from the "cold start" problem, where it is difficult to recommend items to new users who have no history of preferences.
Overall, a combination of content-based and CF approaches may be the most effective in generating accurate and personalized recommendations for users.

1.2 Collaborative Filtering

Collaborative filtering (CF) is a powerful method for developing personalized recommender systems because it relies on the behavior of users, such as their ratings and comments, to make recommendations. User-based CF attempts to find a group of people who have similar preferences to the current user, while item-based CF focuses on identifying items that are similar to those that the user has liked in the past. However, CF can suffer from cold start problems when there is not enough data available about a user or an item, leading to poor recommendations. To mitigate these issues, other types of data such as text, images, and tags can be used to improve the accuracy of recommendations. By leveraging the preferences of similar users, CF can offer new and relevant items to users and help them discover new content that they might not have found otherwise. In a typical CF setup, there would be m user lists U= $\{u_1, u_2, \dots, u_m\}$ and n item lists. $I = \{i_1, i_2, \dots, i_n\}$. The set of things on which each user u i has commented is designated by I_{u_i}. An open rating, usually a numerical scale, is one way in which users might express their thoughts, but ratings can also be derived inferentially from user behavior such as purchasing behavior, time spent on site, and even connecting behavior. Note that if anyone write that $I_{u_i} \subseteq I$, then I_{u_i} might be a collection of zero elements. For a key user, $U_a \subseteq I$ also referred to as the active agent, the goal of a CF process is to determine an item's probability, and this likelihood may take one of two forms [8].

Recommendation: It provides a list of the top N hotels according to the user's overall satisfaction with each establishment $I_r \subset I$. $I_r \subseteq I_{u_a} = \phi$ must include on the suggested list both things and hotels that the present consumer has not previously bought. In certain circles, it is also referred to as a Top-N recommendation method interface [9].

Prediction: $P_{a,j}$ is a numeric number that reflects the anticipated possibility of the item $i_j \notin I_{u_a}$ for the active user u_a, depending on the user's behavior. This value is expressed as a percentage, and it may range from 0 to 1. The information that was supplied by u_a indicates that this forecasted number is somewhere within the same range of 1 to 5 [8].

The CF procedure is shown in Figure 2. the CF procedure starts with inputting data, such as user preferences and item characteristics, and then setting parameters to define the neighbors, which are other users or items that are similar to the current user or item. The CF algorithm then uses this information to make recommendations for new items that the user might like based on the preferences of similar users or the characteristics of similar items. The output of the CF algorithm is a list of recommended items that are ranked according to the likelihood of the user liking them. This procedure can be repeated and refined over time as more data is collected and the algorithm becomes more accurate in predicting user preferences.

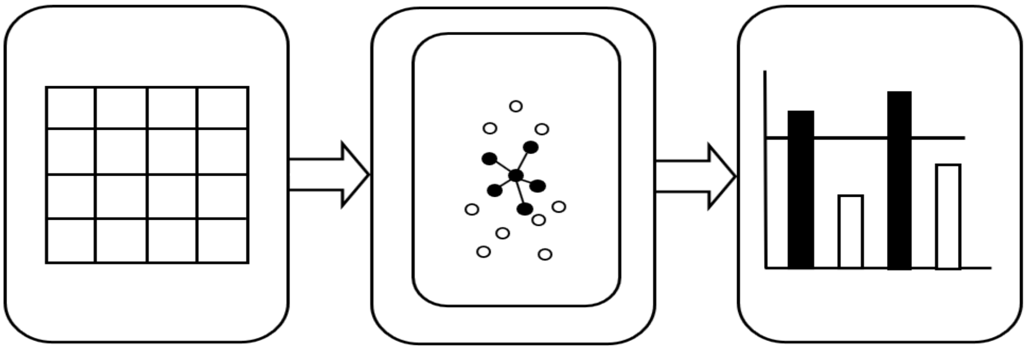

Figure 2. The CF process [9].

2. Related Work

In this section some related work based on a recommendation system using a collaborative filter is discussed below:

Mohd Sabri and Nurul, [10] proposed the item-based CF method was successfully used to create a book recommendation system with an F-measure value of 80.38%. The use of metrics such as Precision, Recall, and F-measure is important in evaluating the performance of such recommendation systems. It's also good to know that the dataset used was rigorously validated through the 10-fold cross-validation method, and that a random selection of 1000 data points was used for training and testing purposes. Overall, the results of this investigation suggest that CF is an effective strategy for building recommendation systems, particularly for book suggestions.

Han et. al., [11] proposed an algorithm that uses triangular fuzzy numbers to improve RSs' ability to understand human emotions, and that it incorporates a layered structure to subjectively perceive similarities across user qualities. It's also noteworthy that the MFPS algorithm outperformed other competitor baselines in terms of effectiveness, explicability, and consistency of the suggestions it provided. It would be great to see more research in this area to further improve the performance of RSs and provide users with even more personalized recommendations.

Sharma et. al., [12] developed a hybrid filtering method in recommender systems, combining both content-based and collaborative filtering approaches. This study suggest that their proposed hybrid filtering system outperforms traditional CF and content-based filtering techniques, highlighting the potential benefits of combining multiple methods for recommendation.

Zhao [13] proposed a novel approach to combine traditional CF recommendation system with a subsystem based on cluster analysis using a genetic algorithm. The proposed system searches for the nearest neighbors only among users who belong to the same cluster in the user base. As a result, the system becomes a genetic clustering-based CF recommender system. The study demonstrated that the response time of a traditional CF recommendation system increases linearly with the number of users, whereas the response time of the proposed genetic clustering-based CF recommendation system remains constant regardless of the size of the user base. The results of the experiments showed that the proposed system outperformed the traditional CF recommendation system in terms of accuracy and response time.

Tahira et. al., [14] have developed a recommender system that considers online customer evaluations and uses aspect-based sentiment classification to match the characteristics of a product that are important to the buyer. The algorithm determines which aspects of the product are most significant to the consumer, finds those elements in customer reviews, and assigns a sentiment score to each feature. The credibility of each customer review determines its weight. They plan to investigate how the algorithm's impact changes with hedonistic and utilitarian items, as the influence of recommender systems depends on the product's nature.

Chen et. al., [15] proposed he DCCF-DN algorithm combines collaborative filtering with dynamic clustering and a double-layer network. The algorithm models user and item attributes as nodes in a double-layer network and uses a rounding-forgetting function to simulate user interest in items. The algorithm then groups users and items into distinct communities using clustering equations and calculates similarities within each community to identify nodes with similar interests. Finally, expected scores are obtained, and a list of top N recommendations is generated. The algorithm has been shown to improve recommendation performance compared to other algorithms in various studies.

Lai et. al., [16] proposed an approach seems interesting and innovative, especially the use of attention-based gated recurrent units for extracting feature aspects from user preferences. The integration of Latent Dirichlet Allocation with attention weights of selected words to produce aspect-based attention semantic vectors is also noteworthy. The use of XGBoost to predict user preference ratings using these vectors is an interesting combination of deep learning and machine learning techniques. It is encouraging to see that the suggested strategy has shown improved prediction accuracy compared to conventional methods.

Bi et. al., [17] The proposed recommendation system based on deep neural networks is designed to address data sparsity and cold-start problems that could arise in collaborative filtering algorithms. It uses various features such as user basic data, item basic data, and item average rating, along with deep neural networks to construct a regression model that can predict user evaluations. The system employs four separate types of neural network layers to create user and item feature matrices using user and item data. Experimental results show that the proposed method outperforms state-of-the-art CF recommendation algorithms and can effectively address data sparsity and cold-start issues.

3. Background Study

Sentiment analysis is a technique that allows us to understand the emotional tone of a piece of text. In the context of recommender systems, sentiment analysis can be used to extract insights from user reviews and incorporate them into the recommendation process. This can help improve the accuracy of recommendations and address issues such as data sparsity and the cold-start problem.

In the study mentioned, sentiment analysis was used to analyze reviews of novels on the Amazon website, and the results were incorporated into a collaborative filtering (CF) recommendation system. An ensemble of models and a weighted vote classifier approach were used to create the system, which showed significant gains in effectiveness when compared to a system without sentiment analysis. The study demonstrates the potential benefits of incorporating sentiment analysis into recommender systems, especially for review-based systems. The approach used in this study is to collect user feedback in the form of reviews from the Amazon website, and then use sentiment analysis to extract useful information from these reviews. An ensemble of models is used for this purpose, and a weighted vote classifier is used for ensemble modeling. Java Web Crawlers are used to scrape the necessary information from the Amazon website. The study focuses on book recommendations, using a dataset of 7210 reviews of 221 novels. Several approaches are used for sentiment analysis, including text normalization and ensemble techniques. The study finds that incorporating sentiment analysis into recommender systems leads to significant gains in their effectiveness, as users are more likely to suggest popular items when sentiment analysis is used

.

4. Problem Formulation

To overcome this challenge, recommender systems are being developed to help users make informed decisions when purchasing products online. By analyzing user data and product information, these systems can make personalized recommendations to users based on their preferences and behavior. This can save users time and effort in searching for relevant products and increase their satisfaction with their purchases. Additionally, recommender systems can help online retailers increase their profits by promoting relevant products to customers and encouraging them to make purchases. Overall, recommender systems are becoming an increasingly important tool for both users and online retailers in the digital age.

5. Research Objectives

- To study and evaluate previous studies on recommendation systems for consumer decisions, researchers have used various approaches. One common approach is to conduct a literature review, where existing studies on the topic are systematically identified, analyzed, and synthesized. Another approach is to perform empirical evaluations of existing recommendation systems, which involves testing their performance using real-world data and comparing it to that of other systems. This approach allows researchers to identify the strengths and weaknesses of different recommendation approaches and make recommendations for improving them. Additionally, researchers can also conduct surveys or focus groups to gather user feedback on their experience with recommendation systems and their preferences for different types of recommendations. By combining these different approaches, researchers can gain a comprehensive understanding of recommendation systems and provide insights on how they can be improved to better serve consumers.
- To create a novel recommendation technique by implementing a new formula for better consumer decisions by using CF.
- To prove the robustness of the proposed model by comparing it with another conventional model in terms of accuracy and other performance evaluation parameters.

6. Research Methodology

the design of a research methodology is important for ensuring the accuracy and reliability of research results. It involves a systematic approach to planning and conducting a study, taking into account factors such as data sources, data analysis techniques, and potential limitations of the study. In the case of developing a novel recommendation system, a well-designed research methodology would be essential to ensuring that the new system is effective and reliable in making better consumer decisions. The use of ensembling models is one approach that can be used to improve the accuracy and effectiveness of recommendation systems.

6.1 Technique Used

In this section, a brief description of all the techniques which are taken into consideration is given below:

6.1.1 Ensemble Classifiers:

Ensemble methods have become popular in various fields, including machine learning and data mining, due to their ability to improve prediction accuracy and reduce overfitting. The method combines several weaker models to form a stronger model that can make more accurate predictions. In the case of sentiment analysis, ensemble methods can be used to classify customer reviews as positive, negative, or neutral.

In the study mentioned earlier, multiple classifiers such as Random Forest, Support Vector Machine, and Logistic Regression were used to build the ensemble model. These models were used in the predicting phase of supervised learning to classify customer reviews based on sentiment. The combination of these classifiers helped to reduce bias, variance and improve prediction accuracy. Overall, ensemble methods have proven to be a useful tool in machine learning and can be applied to various fields, including sentiment analysis and recommendation systems.

- **Random Forest**

Once a random subset of features is selected, the tree is grown using a top-down recursive approach. At each node of the tree, the algorithm selects the feature that best separates the data into the classes based on a criterion such as information gain or Gini index. The process is repeated until the leaf nodes are reached, which contain the final classifications. This process is repeated for each tree in the forest, and the final classification is determined by aggregating the predictions of all the trees.[19]

Random Forest is a popular ensemble method for classification tasks, as it is less prone to overfitting than a single decision tree and can handle large datasets with high dimensionality. It is also computationally efficient, as the trees can be trained in parallel. in the random forest algorithm, the number of input variables is denoted by M, and at each node, a random subset of m variables (where m<<M) is selected to find the best split. The value of m is kept constant during the growth of the forest. The trees in the random forest are unpruned, meaning that they are allowed to grow to their maximum potential size. The final prediction of the random forest is made by aggregating the predictions of all the individual trees, using majority voting for classification and averaging for regression [19].

Random Forest is a popular ensemble learning method that combines multiple decision trees to improve the accuracy and stability of the final classification model. In a random forest, each tree is constructed using a random subset of features from the training data, and the final prediction is made by taking a majority vote of the predictions from all the individual trees [19].

Compared to single tree classifiers like C4.5, Random Forest often performs better in terms of generalization error rate and robustness to noise. Figure 3 shows a block diagram of the Random Forest algorithm, which involves randomly selecting a subset of features and examples to build each decision tree, and then aggregating the results of all the trees to make the final prediction.

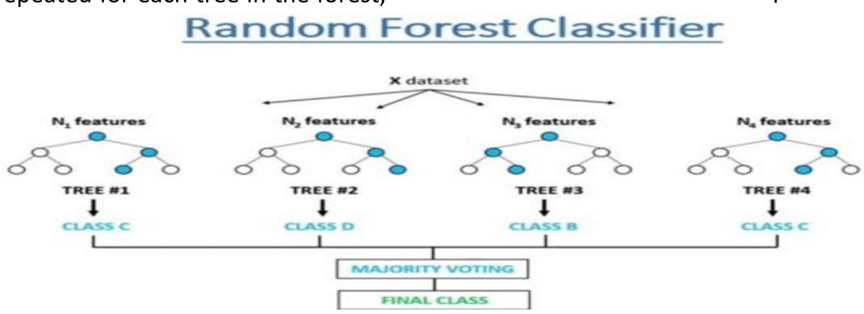

Figure 3. Random Forest [20].

- **Support Vector Machine**

Support Vector Machines (SVMs) are a type of supervised learning algorithm that can be used for classification or regression analysis. SVMs are particularly useful when dealing with complex datasets that have a non-linear relationship between the input variables and the target variable. SVMs work by finding the best hyperplane that separates the data into different classes. The hyperplane is chosen to maximize the margin between the classes, which is the distance between the hyperplane and the nearest data points from each class. SVMs can be extended to non-linear problems by projecting the input vector into a higher-dimensional space using a kernel function [21].

Figure 4 shows an example of an SVM trained on instances from two classes. The maximum margin hyperplane is the dashed line, which separates the two classes with the largest possible margin. The support vectors are the data points closest to the hyperplane, and they define the margin.

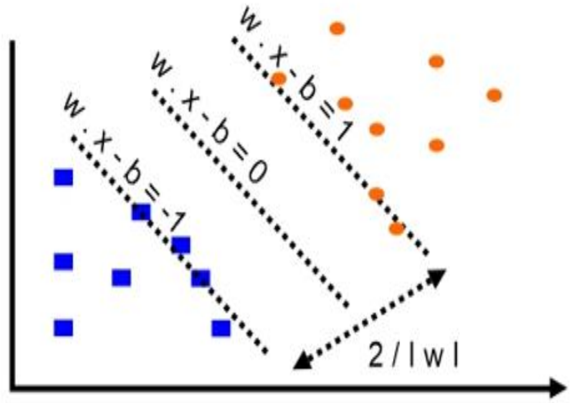

Figure 4. SVM [21].

- **Logistic Regression**

The LR method is widely used for linear classification. It enables the formation of a multivariate regression by allowing a connection to arise between a variable that is independent and dependent variables. LR is the framework of multivariate analysis that may be used to forecast the existence or absence of a function or consequences based on the values of several different predictor variables in a series. This can be helpful in several different contexts [22].

$$\text{Log} = \left[\frac{p}{1-p}\right] = \beta_0 + \beta(\text{Age}) \quad (1)$$

Where p represents the probability, and β_0 indicates the value of the intercept. With the assistance of a line of regression, the LR just splits the data into two distinct categories, as seen in Figure 5 [22].

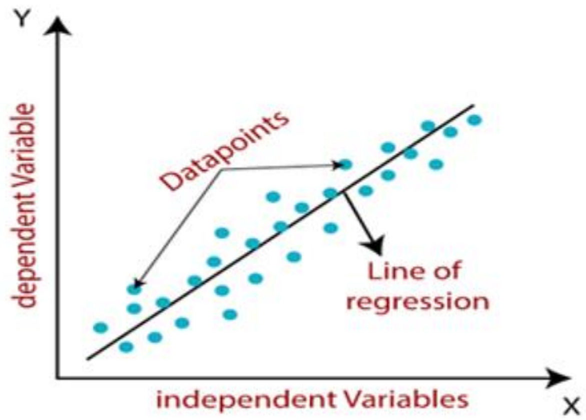

Figure 5. Logistics Regression [22].

6.2 Proposed Methodology

The architecture of the proposed research work has been shown in figure 6 and the work process of this methodology has been given in the below steps.

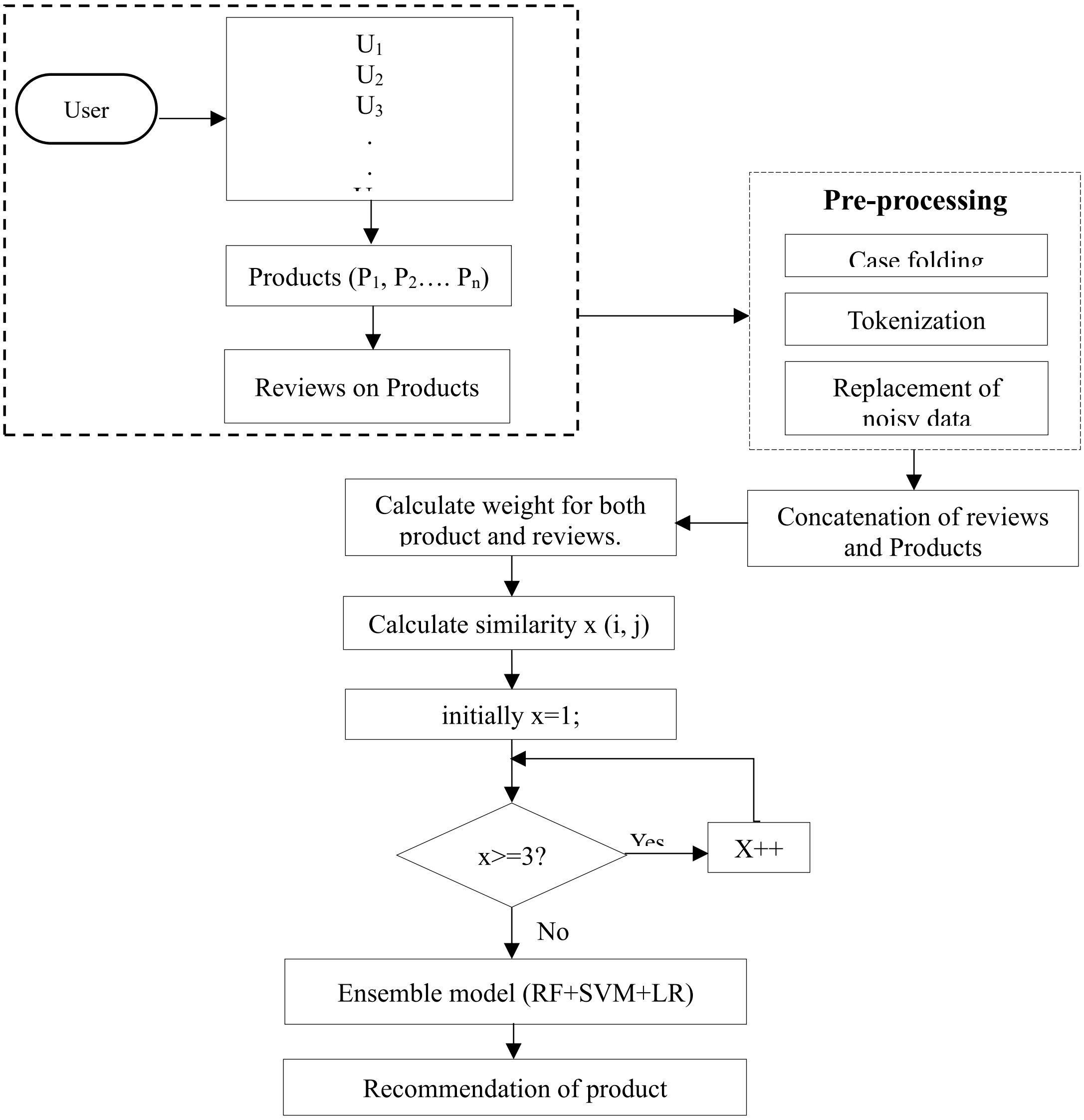

Figure 6. Block Diagram of Proposed Methodology

Step 1: Data Collection

Data collection is the first step of this work. In this step, data is collected from the websites in terms of users (U_1, U_2, U_3, … U_n), products (P_1, P_2, P_3, … P_n), and reviews of the users. The reviews which are taken into consideration are given for the same products by all the users or customers.

Step 2: Pre-processing of the Data

Pre-processing of data is a crucial step in any data analysis or machine learning project, as it helps to clean and transform the raw data into a format that can be easily analyzed by algorithms. Here are some more details on the pre-processing techniques you mentioned:

Case folding: This involves converting all text data to lowercase or uppercase to ensure consistency in the text. This is important because some machine learning algorithms may interpret "Apple" and "apple" as different words, leading to inaccurate results.

Tokenization: This is the process of breaking down text data into individual words or phrases, which are referred to as tokens. Tokenization is a crucial step in text analysis because it allows algorithms to work with individual words or phrases rather than treating entire documents as a single entity.

Replacement of noisy data: Noisy data refers to data that contains errors or inconsistencies, such as misspellings or incomplete records. In order to improve the accuracy of the analysis, noisy data can be replaced or removed entirely. This process is known as data cleaning.

Overall, pre-processing is an important step that can greatly impact the accuracy and effectiveness of machine learning algorithms.

Step 3: Concatenation of the Reviews and Products

In this step 3, Concatenation of the reviews and products refers to the process of merging or combining the reviews of a product with the information about that product. This is typically done by linking each review to the product it pertains to, usually through a unique identifier such as a product ID. The purpose of this step is to create a comprehensive dataset that can be used for analysis and modelling. By combining the reviews and product information, it becomes possible to study the relationship between product features and user opinions or sentiments. This can be useful for identifying key factors that contribute to product satisfaction or dissatisfaction, and for developing strategies to improve product quality or user experience.

Step 4: Calculate the Weight for both Product and Review

After done concatenation of the product and reviews in the previous step, here weight is calculated for products and reviews of the user by using a newly created formula as given below:

$$W^{P}_{(u_p)} = R_{P_{U_i}} / \sum_{U_i \in U}^{n} R_P, \quad (2)$$

Where,

$W^{P}_{(u_p)}$= Denotes the total weight of the product (P) and review by users

R_P = Review (R) of Product given by user (U_i) where i = 1, 2, 3, … n

Step 5: Calculate the Similarity of users' reviews.

In this step 5, the similarity of users' reviews is calculated by using a formula as given below:

$$Sim\,U(i,j) = (1-d) + d \sum_{U_j \in U} Sim\,(U_j) \times R_{P_{U_i}} \quad (3)$$

Where,

$Sim\,U(i,j)$= Denotes the similarity of two consumers' reviews on the same products

d = Dampening factor

Step 6: Ensemble Model

Ensemble models are applied to the data which is obtained in step 5 by calculating similarities of users' reviews. RF, SVM, and LR are employed for modelling with the help of a few classifier algorithms, which collaborate to assign labels to the feedback.

Step 7: Recommendation of products

This is the last step of the whole process. After completing all the processes products are recommended to the consumers as per their needs.

6.3 Proposed Algorithm

ALGORITHM: CONSUMER DECISION USING CF		
Start		
INPUT: →	**PRODUCT:** $P_{i \epsilon 1,2,\ldots n}$	
	CONSUMER: $U_{j \epsilon 1,2,\ldots n}$	User per review
CO-RELATION: →	⟦PRODUCT ↔ CONSUMER⟧	$P_{1,2,3,\ldots U_{i \neq j}}$
OUTPUT: →	PRODUCT ADVOCACY based on TOP-RATED REVIEWS.	

Phase – I: Data Collection

Step 1: Data collection should be accurate based on consumer reviews [U_j] based on each product [P_i].

Step 2: Data contains each product's [P_i] review, available on every e-commerce website.

Phase – II: Data Pre-processing

Step 3: The review for each product and its conciseness will be in Text format. So, it is a Text classification problem.

Step 4: Each review will be cleaned with punctuation, escape sequence, stop words, emoji, unwanted spaces, and digits, then apply WordNet Lemmatization.

Step 5: Each review's conciseness will be converted into a numerical value with either OneHotEncoding.

Step 6: The rating will be visualized with a histogram plot.

Phase – III: Concatenated Review with weight Calculation

Step 7: Each product's threshold will be a minimum with three ratings.

Step 8: Each product's review's conciseness will be considered the top two most frequent ratings.

Step 9: The product will be concatenated with its topmost reviews.

Step 10: To determine each product's review-dominancy, TERM FREQUENCY AND INVERSE DOCUMENT FREQUENCY (TF-IDF) will be calculated to understand the context of the corpus, which can be calculated as:

$$W^{P}_{(u_R)} = \frac{R_{P_{U_i}}}{\sum_{U_i \in U}^{n} R_P}$$

where,

$W^{P}_{(u_R)}$ → denotes the total Weight of the product (P_i) per user's review.

R_P → Review (R) of the Product given by the user (U_i)

Phase – IV: Interlinked between more than two webpages

Step 11: The maximum occurrence of n-gram tokens for each product's review will be considered a dominant specification.

Step 12: For each product [P_i] minimum of three reviews ($R_{P_{U_j}}$) will be extracted to find the occurrence of the most frequent tokens. The damping factor, d, is the likelihood that a user will click on a link, and (1-d) is for non-direct connections to any webpage.

$$Sim\ (i,j) = (1-d) + d \sum_{U_j \in U} Sim\ (U_j) \times W^{P}_{(u_p)}$$

where,

$Sim\ U(i,j)$ → similarity of two consumers' product review

d = Dampening factor

Step 13: The Page Ranking Algorithm is helpful since it ranks webpages by significance, but it is limited because it ranks at indexing time, not retrieval time, and if a webpage with no outlinks is detected, the user goes to a random bookmark.

Phase – V: Ensemble algorithms to find the best algorithm

Step 14: Ensemble Methods build sentiment analysis models. Ensembles combine classifiers to improve outcomes. RANDOM FOREST (**RF**), SUPPORT VECTOR MACHINE (**SVM**), and LOGISTIC REGRESSION(**LR**) are used.

Step 15: Provide classification report.

Step 16: Calculate the best **A**ccuracy, **P**recision, **F**1 – **S**core, and **R**ecall for the best algorithm.

End

---

7. Results and Discussion

In this section, the result demonstrated that are generated based on the proposed methodology. Also, there is a brief explanation of the dataset that is used for the training and testing of the model. Finally, the proposed model is compared with another conventional model to investigate its efficiency of it.

- **Dataset**

The dataset that is used in the proposed methodology is known as Amazon Products Recommendation 2016-2017. It is an open-source dataset that is easily available on the website of Kaggle. It is a vast dataset that Amazon published in 2017 that can be used for computer vision applications such as instance segmentation, object identification, key point recognition, and semantic segmentation. It is a collection of 34661 reviews by the same number of consumers as reviews. In this data, these reviews given on various electronic devices which are sold by amazon are given by the consumers. These devices are two brands such as Amazon and Amazon digital services [23].

Evaluating the performance of a recommender system is crucial to ensure that it is accurate and reliable. There are several commonly used metrics to measure the performance of a recommender system, including:

Accuracy: This metric measures the proportion of correct recommendations made by the system. It is calculated as the number of correct recommendations divided by the total number of recommendations.

Precision: This metric measures the proportion of recommended items that are relevant to the user. It is calculated as the number of relevant recommendations divided by the total number of recommendations.

Recall: This metric measures the proportion of relevant items that are recommended to the user. It is calculated as the number of relevant recommendations divided by the total number of relevant items.

Error rate: This metric measures the average difference between the predicted and actual ratings of the recommended items.

F-measure: This metric is used to balance the precision and recall of the recommender system. It is calculated as the harmonic mean of precision and recall.

By evaluating the performance of a recommender system using these metrics, we can identify areas for improvement and fine-tune the system to make better recommendations to users. Precision is calculated by;

$$Precision = \frac{TP}{(TP + FP)} \quad (4)$$

Where,

TP = True positive value,

FP = False Positive value.

Recall obtained by using.

$$Recall = \frac{TP}{(TP + FN)} \quad (5)$$

Where,

FN = False Negative

The F1-score measured by using the formula is;

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \quad (6)$$

To calculate the accuracy of the model a formula is used which is given below;

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

The results which are calculated for the proposed methods are given in R1, R2, R3, R4, and R5 as shown below.

R1: RF

This result presents the results of calculating the precision, recall, F1-measure, and accuracy of the proposed RF for the products and reviews that are taken from the dataset. The values of the specified parameters that are computed for the RF in this work are shown in table 2 below. A graph representation of this result is shown in figure 7 which is given below table 1.

Table 1. Calculated Values of parameters for RF

Technique	Precision	Recall	F1-Measure	Accuracy
RF	0.791	0.78	0.783	0.799

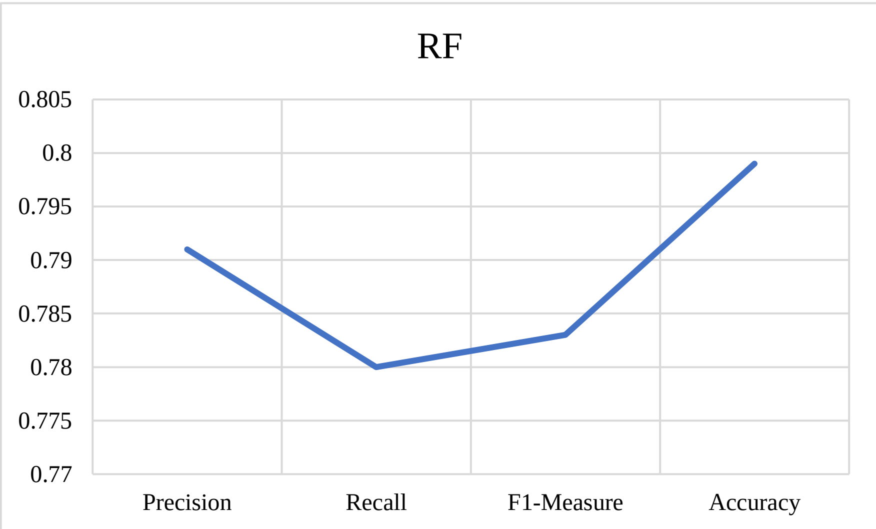

Figure 7. Graph of RF Results

R2: SVM

The value of the precision, recall, f 1-measure, and accuracy are calculated for the proposed SVM in this result in the context of reviews and products. The table shows the values of the parameters which are calculated for the SVM in the work are shown in table 2 and the graph representation of this result is shown in figure 8 as given below.

Table 2. Calculated Values of parameters for SVM

Technique	Precision	Recall	F1-Measure	Accuracy
SVM	0.775	0.761	0.762	0.785

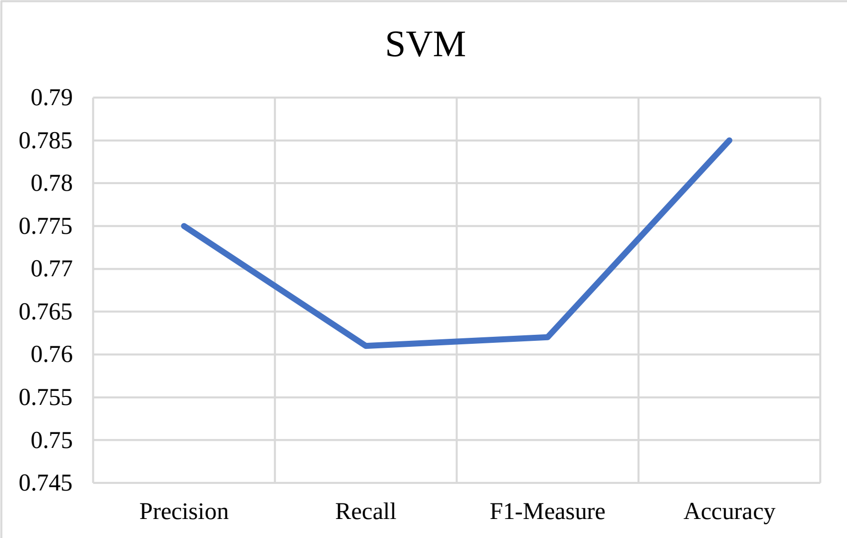

Figure 8. Graph of SVM results

R3: LR

In the context of reviews and products, the value of the precision, recall, f 1-measure, and accuracy are determined for the suggested LR in this result. The values of the parameters that are computed for the LR in this work are displayed in table 3, and a graph representation of this result is shown in figure 9 which is presented below.

Table3. Calculated Values of parameters for LR

Technique	Precision	Recall	F1-Measure	Accuracy
LR	0.798	0.791	0.781	0.811

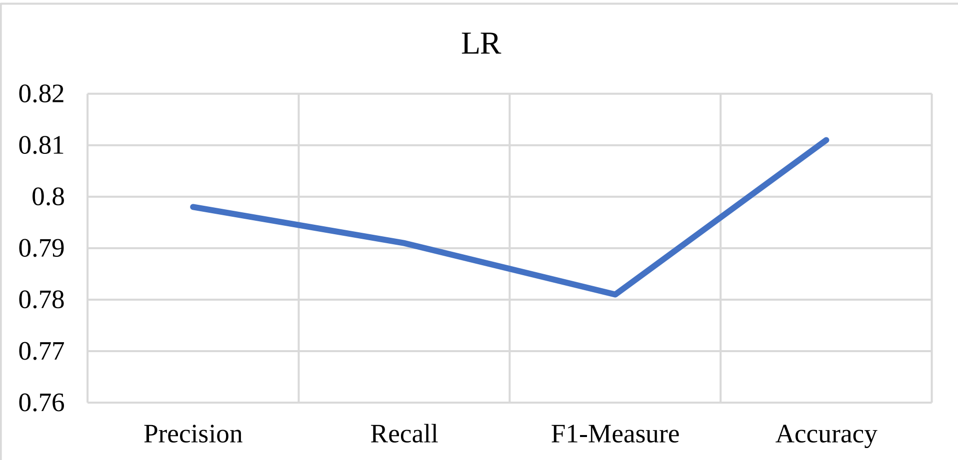

Figure 9. Graph of LR results

R4: Ensemble Classifiers

This result is obtained for ensemble classifiers. RF, SVM, and LR are combined in an ensemble model to obtain better results in comparison to the previous research. In the context of reviews and products, the value of the precision, recall, f1-measure, and accuracy are determined for the suggested ensemble classifiers in this result. The work presents table 4 that provides the values of the parameters that are computed for the ensemble model, and it presents figure 10 that shows a graph representation of the result.

Table 4. Calculated Values of parameters for Ensemble Classifiers

Technique	Parameters			
	Precision	Recall	F1-Measure	Accuracy
Ensemble Classifiers	0.811	0.798	0.809	0.821

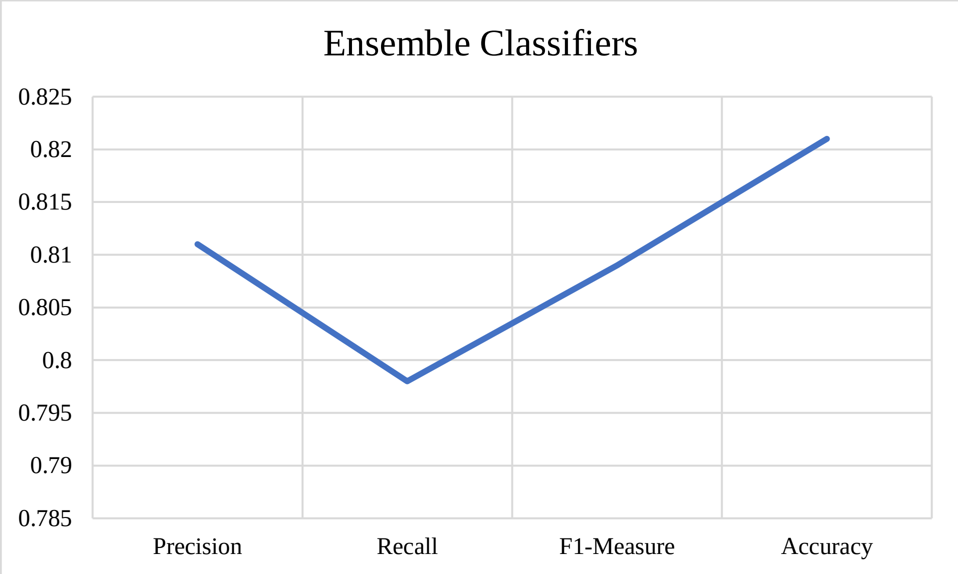

Figure 10. Graph of Ensemble Classifier results

R5: Error rate of all techniques

In this result, firstly the error rate of all techniques is calculated separately. Then, the error rate of the ensemble model (RF+SVM+LR) is calculated to get a lower error than the previous work. The error rate of all these techniques is depicted in table 5 and a graph representation of this result is shown in figure 11 as given below.

Table 5. Calculated Values of Error Rate of all the techniques

Technique	Error Rate
RF	0.108
SVM	0.153
LR	0.1097
Ensemble Classifiers (RF+SVM+LR)	0.100

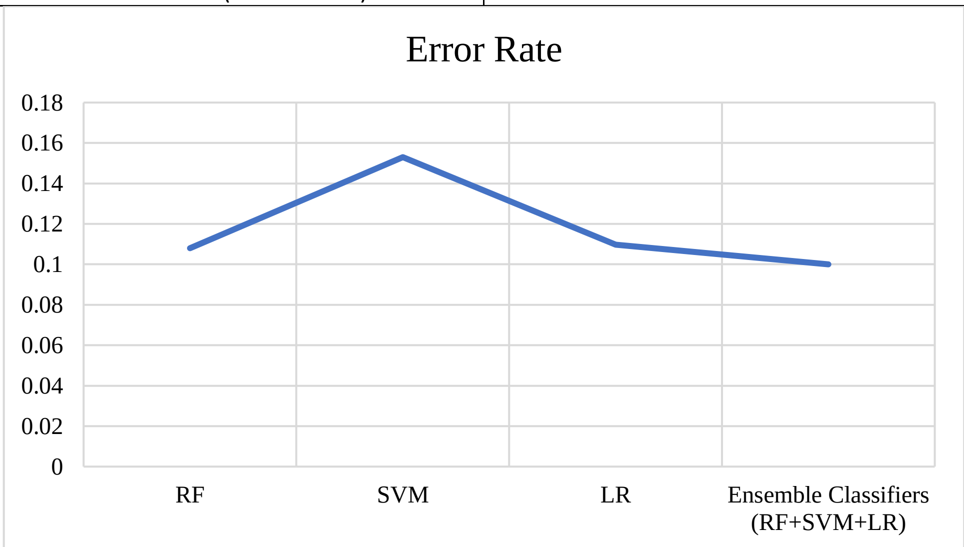

Figure 11. Graph of Error rate results for all techniques

These results were obtained by repeating the model for each classifier 3 times. It was revealed that the ensemble model with an accuracy rate of 95.2% had higher accuracy than the other algorithms for modelling.

8. Comparative Analysis

In this section, the proposed model is compared with other conventional methods such as RF, SVM, and LR. It is compared based on positive metrics parameters such as accuracy, precision, recall, f1-measure, and error rate. Figure 12 shows the comparison of the conventional technique with the proposed model based on precision, recall, f1-measure, and accuracy and it is seen the proposed model is higher among all the methods. Figure 13 shows the comparison of the conventional technique with the proposed model based on error rate and it is seen the proposed model is a lower error rate among all the methods. Table 6 shows the overall comparison of the proposed model with other conventional techniques in terms of precision, recall, f1-measure, and accuracy, and table 6 shows the overall comparison of the proposed model with other conventional techniques in terms of error rate.

Table 6. Comparison of Results

Parameters	Models						
	MNB [18]	MLP [18]	LR [18]	RF	SVM	LR	Proposed Ensemble Classifiers (RF+SVM+LR)
Precision	0.783	0.753	0.783	0.798	0.775	0.798	0.802
Recall	0.774	0.748	0.774	0.791	0.761	0.791	0.821
F1-measure	0.777	0.747	0.777	0.821	0.762	0.781	0.833
Accuracy	0.7907	0.7599	0.7911	0.925	0.785	0.811	0.821

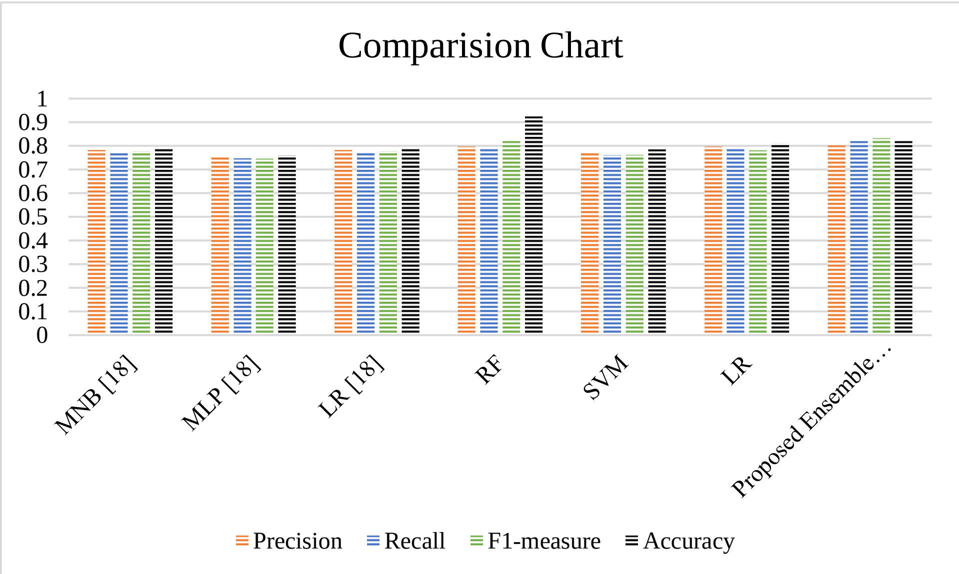

Figure 12. Comparison of the proposed work's performance to that of similar current schemes in terms of Precision, Recall, F1-measure, and Accuracy

Table 7. Error Rate

Parameter	Models						
	MNB [18]	MLP [18]	LR [18]	RF	SVM	LR	Proposed Ensemble Classifiers (RF+SVM+LR)
Error Rate	0.207	0.239	0.208	0.108	0.153	0.1097	0.100

Figure 13. Comparison graph of Error rate among all techniques

9. Conclusion and Future Work

This work focuses mainly on proposing novel and group trust models to enhance consumer decisions. Recommendation systems have been and continue to be applied in various applications to support item (e.g., movies or music) recommendation and to solve the information overload problem by suggesting items of possible interest to consumers. In the work, CF is used to create a novel weighted recommendation system to better consumer decisions. Most recommendation systems mainly depend on a method known as CF, in which the customer's preference data is joined with that of other users to produce predictions about further items that the consumer may be interested in. Using CF, a unique weighted recommendation system has been developed in this study to improve consumer choice-making. The technique includes the creation of a formula to compute the weight of both the product and the review, as well as a calculation to assess the similarity between different consumers' reviews. In the approach, RF, SVM, and LR are the components that make up the ensemble model. Ensemble Classifiers, consisting of RF, SVM, and LR, are considered when putting the findings of the approach into practice to get superior outcomes compared to those obtained from earlier research. The suggested model is trained and evaluated with the use of a publicly accessible open-source dataset that can be found on the Kaggle website. In addition to that, a comparison of the outcomes is included in the study. According to the results of a numerical study of the suggested model, it outperformed other traditional approaches in many different respects, including accuracy (0.821), precision (0.802), recall (0.821), F-measure (0.833), and error rate (0.100). In further work, other algorithms would be evaluated for performance comparison to determine the recommendation system that is the most effective, particularly for online shopping websites.

References

[1]. Cui, Laizhong, Linyong Dong, Xianghua Fu, Zhenkun Wen, Nan Lu, and Guanjing Zhang. "A video recommendation algorithm based on the combination of video content and social network." Concurrency and Computation: Practice and Experience 29, no. 14 (2017): e3900.

[2]. Liu, Mengsi, Weike Pan, Miao Liu, Yaofeng Chen, Xiaogang Peng, and Zhong Ming. "Mixed similarity learning for recommendation with implicit feedback." Knowledge-Based Systems 119 (2017): 178-185.

[3]. Ding, Linlin, Baishuo Han, Shu Wang, Xiaoguang Li, and Baoyan Song. "User-centered recommendation using us-elm based on dynamic graph model in e-commerce." International Journal of Machine Learning and Cybernetics 10 (2019): 693-703.

[4]. Bouras, Christos, and Vassilis Tsogkas. "Improving news articles recommendations via user clustering." International Journal of Machine Learning and Cybernetics 8 (2017): 223-237.

[5]. Kim, Byung-Do, and Sun-Ok Kim. "A new recommender system to combine content-based and collaborative filtering systems." Journal of Database Marketing & Customer Strategy Management 8 (2001): 244-252.

[6]. Sivaramakrishnan, N., and V. Subramaniyaswamy. "GPU-based Collaborative Filtering Recommendation System using Task parallelism approach." 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics, and Cloud)(I-SMAC), 2018 2nd International Conference on. IEEE, 2018

[7]. Deng, Xiaoyi, Fuzhen Zhuang, and Zhiguo Zhu. "Neural variational collaborative filtering with side information for a top-K recommendation." International Journal of Machine Learning and Cybernetics 10 (2019): 3273-3284.

[8]. He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. "Neural collaborative filtering." In Proceedings of the 26th international conference on world wide web, pp. 173-182. 2017.

[9]. Wang, Wei, et al. "Trust-enhanced collaborative filtering for personalized point of interests recommendation." IEEE Transactions on Industrial Informatics 16.9 (2019): 6124-6132.

[10]. Mohd Sabri, Norlina, and Nurul Azeymasnita Jaffar. "Book recommendation based on collaborative filtering technique." ESTEEM Academic Journal 18 (2022): 92-103.

[11]. Han, Di, Yijun Chen, and Shuya Zhang. "Implicit social recommendation algorithm based on multilayer fuzzy perception similarity." International Journal of Machine Learning and Cybernetics 13, no. 2 (2022): 357-369.

[12]. Sharma, Sunny, Vijay Rana, and Manisha Malhotra. "Automatic recommendation system based on hybrid filtering algorithm." Education and Information Technologies (2022): 1-16.

[13]. Zhao, Yan. "Design of Garment Style Recommendation System Based on Interactive Genetic Algorithm." Computational Intelligence and Neuroscience 2022 (2022).

[14]. Tahira, Anum, Walayat Hussain, and Arif Ali. "Based Recommender System for Hedonic and Utilitarian Products in IoT Framework." In IoT as a Service: 7th EAI International Conference, IoTaaS 2021, Sydney, Australia, December 13–14, 2021, Proceedings, pp. 221-232. Cham: Springer International Publishing, 2022.

[15]. Chen, Jianrui, Bo Wang, Zhiping Ouyang, and Zhihui Wang. "Dynamic clustering collaborative filtering recommendation algorithm based on double-layer network." International Journal of Machine Learning and Cybernetics 12 (2021): 1097-1113.

[16]. Lai, Chin-Hui, Duen-Ren Liu, and Kun-Sin Lien. "A hybrid of XGBoost and aspect-based review mining with attention neural network for user preference prediction." International Journal of Machine Learning and Cybernetics 12 (2021): 1203-1217.

[17]. Bi, Jian-Wu, Yang Liu, and Zhi-Ping Fan. "A deep neural networks-based recommendation algorithm using user and item basic data." International Journal of Machine Learning and Cybernetics 11 (2020): 763-777

[18]. Abbasi, Fatemeh, and Ameneh Khadivar. "Collaborative Filtering Recommendation System through Sentiment Analysis." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, no. 14 (2021): 1843-1853.

[19]. Ali, Jehad, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. "Random forests and decision trees." *International Journal of Computer Science Issues (IJCSI)* 9, no. 5 (2012): 272.

[20]. https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6

[21]. Katzfuss, Matthias, Jonathan R. Stroud, and Christopher K. Wikle. "Understanding the ensemble Kalman filter." The American Statistician 70, no. 4 (2016): 350-357.

[22]. Ribokaitė, Lina. "Outlier detection in multidimensional streaming data." Ph.D. diss., Vilniaus universitetas, 2021.

[23]. https://www.kaggle.com/code/haojie98/amazon-product-recommendations/data+9