

Article

SmartTips: Online Products Recommendations System Based on Analyzing Customers Reviews

Noaman M. Ali ^{1,*} , Abdullah Alshahrani ², Ahmed M. Alghamdi ³  and Boris Novikov ⁴ ¹ Department of Information Systems and Technology, Port Said University, Port Fouad, Port Said 42526, Egypt² Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia³ Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia⁴ Department of Informatics, National Research University Higher School of Economics, 3A Kantemirovskaya St., 194100 Saint Petersburg, Russia

* Correspondence: no3man_mohamed@himc.psu.edu.eg

Abstract: Online customers' opinions represent a significant resource for both customers and enterprises to extract much information that helps them make the right decision. Finding relevant data while searching the internet is a big challenge for web users, known as the "Problem of Information Overload". Recommender systems have been recognized as a promising way of solving such problems. In this paper, a product recommendation system called "SmartTips" is introduced. The proposed model is built based on aspect-based sentiment analysis, which exploits customers' feedback and applies the aspect term extraction model to rate various products and extract user preferences as well. Several factors were considered, including readers' votes, aspect term frequency, opinion words' frequencies, etc. We tested our model on benchmark datasets that are widely used, and the results show that it outperforms the baseline methods regarding the mean squared errors of generated predictions.

Keywords: recommender systems; aspect-based sentiment analysis; collaborative filtering; natural language processing; aspect-term extraction; feature extraction; text analysis



Citation: Ali, N.M.; Alshahrani, A.; Alghamdi, A.M.; Novikov, B. SmartTips: Online Products Recommendations System Based on Analyzing Customers Reviews. *Appl. Sci.* **2022**, *12*, 8823. <https://doi.org/10.3390/app12178823>

Academic Editor: Valentino Santucci

Received: 29 July 2022

Accepted: 27 August 2022

Published: 2 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the diversity and abundance of available alternatives of goods and services represent a double-edged sword for both customers and enterprises. From the customers' side, it provides many options to meet all the different needs of customers, but it adds a heavy burden on them in choosing the most appropriate option. On the other hand, it imposes enormous challenges on companies in obtaining a market share due to intense competition and the difficulty of attracting new customers and ensuring the loyalty of existing customers under the open market system [1].

Commonly, customers' opinions represent a significant resource for both customers and enterprises to extract much information that helps them make the right decision, where companies resort to analyzing this type of data to obtain insights into the market trend and the extent of customer satisfaction with their products and analyze future needs, etc. In contrast, customers who are about to purchase a new product or receive a specific service resort to reading customer remarks to evaluate alternatives and choose the most appropriate choice [2–4]. Practically, it is difficult for customers to read this large amount of reviews, which may take much time and result in missing many of the exceptional marketing offers [5]. Therefore, this work mainly concerns studying the customer side and developing a solution for helping them. Based on the preceding, the need arises to build an intelligent recommender system that exploits the existing customers' experience to help them make appropriate decisions based on their preferences.

Generally, the problem of finding relevant data while searching the internet represents a big challenge for web users due to the enormous amounts of available information on the web. These difficulties are related to the well-known problem of information overload. Recommender systems guide users towards the more interesting or valuable objects in a large area of possible options based on their preferences. It aims to predict users' intentions and data needs to facilitate and customize their online experience and help them by providing a list of suggested proper resources. There is a wide range of recommender systems. Most of them vary in the work base (e.g., web content mining, web usage mining, etc.) [6–8].

This work presents an online product recommendation system called “*SmartTips*” built based on analyzing customer reviews utilizing aspect-based sentiment analysis. It starts with capturing the user preferences using historical review data by extracting aspect terms and measuring the user sentiments towards each aspect separately. Next, the model searches the previously rated product sets for similar products to the current user preferences. Review data are used to rate various products using review text and users' votes as well. Finally, the recommendation engine generates a list of products that may be interesting to the user using a ranking scheme that applies a weight reflecting the relevance of each aspect in accordance with the user preferences.

1.1. Research Objectives

In this work, we strive to accomplish the following primary objectives. First, we aim to develop a more reliable domain-specific recommendation model for handling the problems of information overload and cold start, taking into account the main characteristics of big data affecting the performance of traditional methods. In other words, we aim to develop a scalable framework for online product recommendations with stable performance. Moreover, we aim to find and select diversified datasets suitable for validating the online recommendation system.

1.2. Contributions

In order to fulfill these objectives, we developed an online products recommendation system called “*SmartTips*” capable of addressing the shortcomings of existing models. The proposed model is built based on aspect-based sentiment analysis, where aspect terms were used to rate various products and extract user preferences as well. The proposed domain-specific model handles the problem of cold start where it has guarantees to generate recommendations even if the user has no historical data, i.e., “New User”. The main procedures of the proposed system are Rating Products, Extracting Preferences, and Generating Recommendations. Additionally, to validate and evaluate our proposed system, we used a benchmark dataset that contains ratings and review texts collected from the Amazon website by Julian McAuley and Jure Leskovec (2013) [3]. We evaluated the performance of the proposed model; the obtained results demonstrate its validity and reliability.

The rest of this paper is organized as follows: In Section 2, we briefly present a discussion of the basic and related concepts of recommender systems, including the basic definition, methods for development, and the applications' domains. In Section 3, we briefly present a discussion of the related studies. Section 4 introduces the proposed model with a description of the developed framework; the basic components of our model are described in detail as well. Section 5 provides details of the conducted experiments, including a description of the datasets used in this study. The implementation details and an exploration of the obtained results are provided with a detailed discussion to highlight our work's theoretical and practical implications; the testing mechanism, evaluation metrics, and criteria are described in this context. Finally, the conclusion summarizes the results and future work.

2. Background

This section provides an overview of recommender systems (RSs). It starts with arguing the basic concepts related to RSs, including the definition, purposes, and the standard architecture for building RSs. A comprehensive discussion of the existing approaches for creating recommender systems is subsequent. Moreover, popular applications of recommender systems are discussed.

2.1. Basic Concepts of RSs

The internet represents the primary source that users rely on for information searching and gathering. The volume of information available on the internet is rapidly growing due to the ongoing expansion of the World Wide Web and its tremendous applications. So, the information overload problem represents users' biggest challenge during web searching. This situation occurs when people are flooded with a lot of information and service options. Therefore, obtaining more "relevant" or "interesting" information becomes difficult for users. Recommender systems were proposed and developed to alleviate the problem of information overloading by providing users with a list of recommendations tailored to their preferences [9]. Therefore, recommender systems could be defined as "a system that analyzes a user's data and tries to model his behavior in order to generate a list of ranked objects that could be interesting for a specific user and finally sort these objects and provide him with the top-ranked items". Based on the above definition; such systems could be considered a subclass of information filtering systems. Figure 1 shows the basic pipeline to be followed for building a recommender system. In order to build a recommender system, several data forms could be employed to accomplish the desired task, including historical data (e.g., purchase history) and preferences (e.g., the user profile data and settings); also, data of other users may be employed to compare the current user's behavior to others through measuring the similarities between distinct users' behaviors.

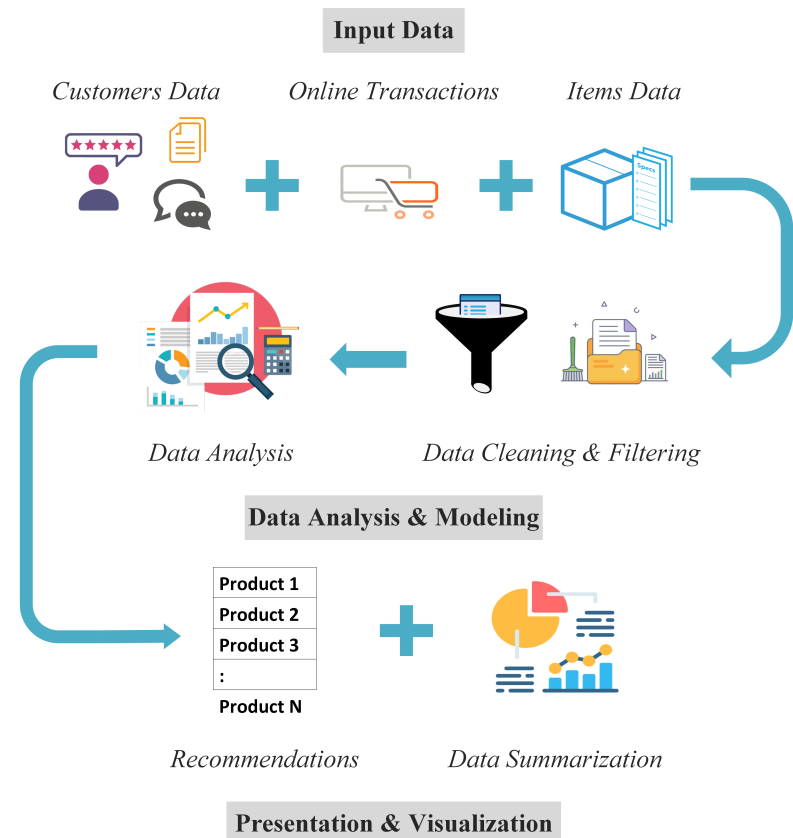


Figure 1. General architecture of a recommender system.

2.2. Methods for Creating RSs

Recently, there has been an increasing interest in developing recommender systems; integrating a recommendation engine has become essential while building or updating any web application and mobile application. The most common classification of approaches for creating a recommender system involves three main categories: Content-based, Collaborative Filtering, and Hybrid recommender systems. Other researchers introduced different classifications, including Context-Aware, Group, Multi-Criteria, Cross-Domain, Multi-Stakeholder, and Multi-Task [10]. Figure 2 shows the hierarchy of the recommender system approach classification, following a brief discussion about these classes.

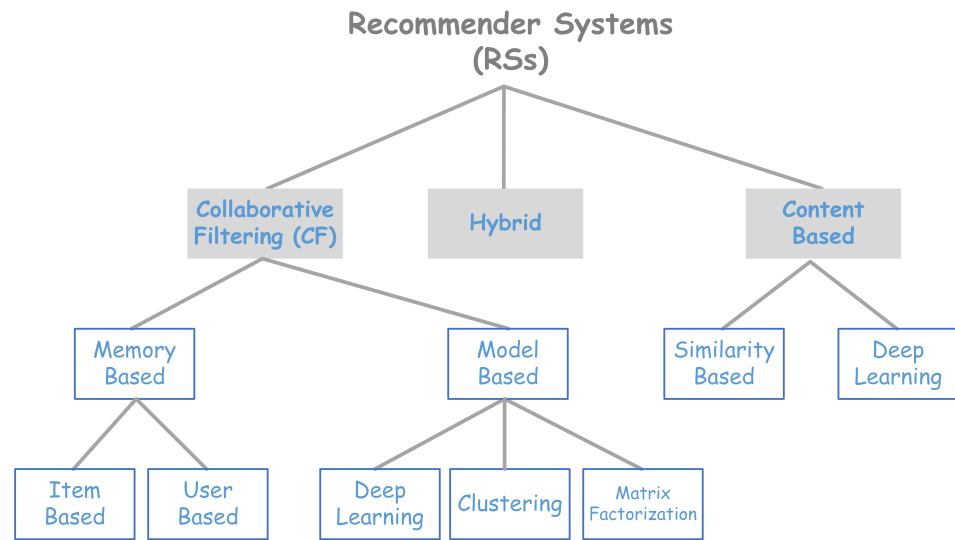


Figure 2. Methods for creating recommender systems.

2.2.1. Content-Based

Content-oriented recommendations are based primarily on creating a profile for each user and subject as well. The underlying technique is to match the user with the object he/she liked (e.g., product and content). In other words, the historical behaviors of the user are used to make recommendations for him/her, including implicit or explicit feedback. Comparisons across users and items are performed to measure their similarities in order to decide which content is more suitable for the user. Vectorization is the primary step to creating users' and items' profiles; it converts objects into vectors by producing Item Vectors and User Vectors. These profiles are then used to measure the similarity between objects. There are several ways to accomplish this task; the most common methods are [11–17].

Cosine Similarity: Employing this method is recommended when the features are high-dimensional, specifically in the areas of document clustering, text analysis, ontology matching, and information retrieval.

Jaccard Similarity: Also known as intersection over union, it is used for object-to-object likeness. The best employment is when the vectors contain binary values; with more than two possible values, the Jaccard affinity does not apply.

2.2.2. Collaborative Filtering

Collaborative-oriented recommendations are based primarily on creating the user–item Utility Matrix. Unlike the content-centric approach, there are no attributes that match users or subjects. The underlying technique is to learn from the historical transaction made by users, including explicit feedback such as users' ratings on different items or implicit feedback such as transactions history (e.g., purchased and browsing). Therefore, the main assumption of collaborative filtering is to use similar user behaviors to make recommendations to a specific user. There are two main subcategories that collaborative filtering can be divided into [18–24].

Memory-based approach: For this approach, the user-item Utility Matrix is remembered, and recommendations are made by querying a given user with the rest of the Utility Matrix. Such techniques neither require performing dimensionality reduction nor model fitting. Commonly, these approaches could be divided into two primary sections: Item–Item filtering and User–User filtering approaches.

Model-driven approach: In this approach, it is not required to remember the Utility Matrix. It tries to model the user’s behavior from the matrix. Here, reducing the dimensionality and fitting the model are essential. The most common implementations of the model-driven approach include the employment of Clustering algorithms, Deep Learning methods, and Matrix-Factorization-based algorithms.

2.2.3. Hybrid Models

A hybrid recommendation system refers to systems that integrate two or more models that belong to different categories: content-based and collaborative filtering approaches. Therefore, it could be considered a special type of recommender system. The primary goal of developing these combinations is to overcome the shortcoming of using participating approaches separately and increase their efficiency. Several implementations of hybrid recommender systems were introduced; one of them uses two selected approaches separately and then combines the predictions. The most common combinations are introduced by Robin Burke that include seven models: Weighted, Switching, Mixed, Feature Combination, Feature Augmentation, Cascade, and Meta-Level [25].

2.3. Applications of RSs

Recently, recommender systems have been applied in several domains to facilitate access to desired information or assist users in making the most suitable decision; the mentioned domains include online streaming, e-commerce, social networks, education, etc. Actually, the applications of recommender systems in e-commerce have acquired the largest share of contributions. The widespread nature of such applications is due to their economic nature and the importance they represent for enterprises and customers as well. Moreover, the advent of mobile technologies plays a significant role. The recommendation engine is becoming an integral part of the web and mobile applications; most of these applications are built using the Machine and Deep Learning techniques [5,26–32]. Several examples of recommender systems in this field include:

- **Online-contents Recommendations:** Offering new content to enhance user engagement to the website. The goal is to introduce users to new content that may attract their interest and motivate them to consume more content; online content may include movies, books, news, etc.
- **Stocks Recommendations:** Present the most profitable stocks to customers. The novelty does not matter, where the list of recommendations may include early used promotions by the user; the importance relies on the return on equity.
- **Products Recommendations:** Recommend a combination of products, including previously consumed and new products. For the first class of products, the history of users’ transactions serves as a reminder of their frequent purchases. On the other hand, new products may be suggested according to similar users’ behavior.
- **Conversational Recommender Systems (CRSs):** It is a tool that tries to help the user determine which products he/she is going to buy by conversational mechanism. This agent can mimic a natural conversation between client and professional sales support to arouse customer preference.

3. Related Works

Recently, the development of recommender systems has become more interactive, drawing increasing interest, and attracting researchers from various scientific disciplines [3,5,10,33]. The availability of a massive volume of review data encourages researchers to develop models that are based on text analysis. Aspect-based sentiment

analysis appears a promising approach for measuring the users' sentiment about specific aspects. Therefore, the development of such systems strives to capture users' preferences from their historical feedback. The primary goal is to provide users with a list of items (e.g., products and service providers) that may be interesting, following a brief review of the recent approaches for developing recommender systems based on the text analysis.

Concerning the product recommendation models, Lin Qiu et al. [9] have introduced online review texts and rating integration. The proposed Aspect-based Latent Factor Model (ALFM) starts with grouping review texts by users and similarly by different items and calculating word frequencies to build a user-review matrix and item-attribute matrix, respectively. It also considers the polarity scores and item-word frequencies of corresponding words. Subsequently, the item latent factors with word latent factors and user latent factors can be derived from the built matrices. The proposed model acts well in the cross-domain task in addition to the rating prediction task through transferring word embedding. The authors in [34] have introduced a computational model for online products recommendations. The proposed conversational recommender system incorporates the product's functional requirements for interactions with users—the working mechanism is based on collecting information from the client by asking questions. The model integrates both the ontology and structure in generating the interactions. The developed system is able to handle multi-function and multi-feature products domains. Moreover, Aminu Da'u et al. [35] have introduced the integration of aspect-based sentiment analysis in generating recommendations. The proposed work incorporates the Deep Learning technique to extract aspects from review data. Additionally, the system generates aspect-oriented ratings through computing the sentiment polarities. Finally, the generated ratings are integrated into a tensor factorization machine in order to compute the overall rating predictions. Similarly, the authors in [36] have proposed a collaborative filtering recommendation model that is based on using aspect-based sentiment analysis. The model incorporates using the multichannel deep convolutional neural network to extract product aspect terms from the review text and then using a weighting scheme in order to rate various aspects and generate an aspects rating matrix. Various aspects' ratings are then integrated to compute the general product rate. Furthermore, the authors in [37] proposed a collaborative filtering recommender system that utilizes the neural-network-based text representation in order to measure similarities between review texts. Seven model combinations were introduced, including lexical-based and Machine-Learning-based approaches to compute the users' text similarity; the Long Short Term Memory (LSTM)-based model has achieved the best performance results.

On the other hand, Elham Asani et al. [38] introduced a restaurant recommender system that analyzes users' feedback and extracts food preferences to generate recommendations based on extracted preferences. The proposed model extracts food names from comments and then clusters them; the sentiment analysis is applied to measure polarity towards various food names. Restaurants' recommendations are made based on similarities between user preferences and restaurants ratings.

Concerning hotel recommendations, Biswarup Ray et al. [39] introduce a hotel recommender system based on sentiment analysis and aspect-based categorization techniques. They also introduced a collected dataset from Tripadvisor.com that involves online hotel reviews. The proposed ensemble starts with creating word embeddings of the clients' review text using the Bidirectional Encoder Representations from Transformers (BERT) model. After that, a weight assigning protocol was applied that merged with three phases for neutral-negative, positive-negative, and neutral-positive sentiments. Finally, recommendations were generated by grouping the reviews into different classes by measuring their similarities using cosine similarity and fuzzy logic.

4. The Proposed System for Product Recommendations

This section presents details of our proposed system, “SmartTips”, which exploits the aspect terms extracted from the text for building the recommender system [40]. Generally, the proposed system consists of three different components: (1) Rating Products, (2) Extracting Preferences, and (3) Generating Recommendations. Figure 3 shows the proposed recommender system workflow that shows the interaction between different components, following a description of these tasks in detail.

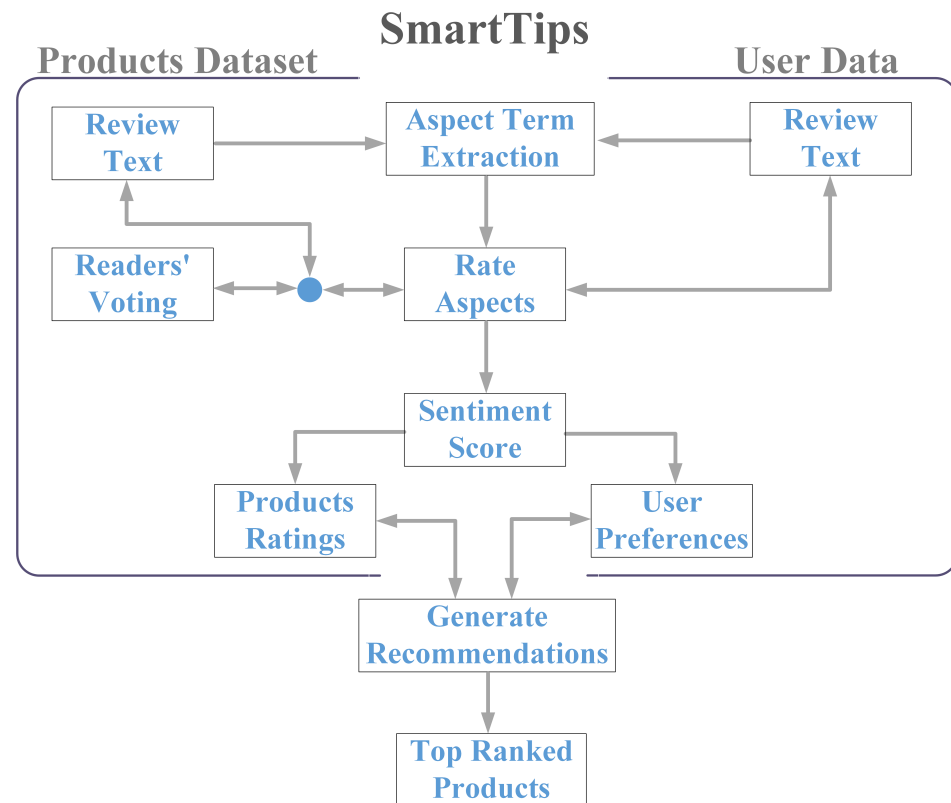


Figure 3. The overall scheme of the proposed recommender system.

4.1. Rating Products

The primary task of the proposed recommender system is rating products, which has been built primarily on the prominent aspect terms within the text [40]. Generally, the proposed model for rating products considers the following information: review text, aspect term frequency, sentiment words, and their polarities, where the number of positive or negative sentiment words towards each aspect term were considered, and readers’ vote on each review containing one or more aspect terms. Moreover, we consider only reviews from actual practice by accepting reviews submitted by customers with verified purchases.

The inputs to this task are the review data, including the review text itself, and meta-data, specifically the readers’ votes. We believe that positive readers’ votes are a significant indicator of the relevance and confidence of the review’s assumptions; conversely, negative votes represent a justification of distrust of the customer’s assumptions. The negative experiences of some customers with a particular product or company may push them to take revenge by writing some imaginary defects in the product, so the readers’ disagreement with the comment is considered evidence of the lack of credibility of the review.

The model starts with applying the aspect term extraction model to the review text—next, the extraction of opinion words associated with aspect terms from each review, respectively. The dependency relations extracted through applying the dependency parser with the help of POS tags is the more suitable method for extracting the opin-

ion words. Subsequently, measuring the polarity of these words using the lexical-based technique to rate the extracted aspects. Algorithm 1 presents the technique followed for computing the sentiment score for each aspect related to a specific product in the dataset.

Algorithm 1 Computing the sentiment score

Input: (LIST: *Review_Data*)

(LIST: *Aspect_Terms*)

Output: (LIST: *Aspect_Ratings*)

Initialization :

```

1: Create  $\leftarrow$  (LIST : Aspect_Ratings)
2: for each review  $\in$  Review_Data do
3:   Extract  $\leftarrow$  review  $\Rightarrow \{v^+, v^-\}$ 
4:   Extract  $\leftarrow$  review  $\Rightarrow$  "review_text"
5:   for each token  $\in$  review_text do
6:     if token  $\in$  Aspect_Terms then
7:       if token  $\notin$  Aspect_Ratings then
8:         Append  $\leftarrow$  token  $\Rightarrow$  "Aspect_Ratings"
9:       end if
10:      Extract  $\leftarrow$  review_text : token  $\Rightarrow$  "sentiment_words"
11:      Count  $\leftarrow$  sentiment_words  $\Rightarrow$  "csw"
12:      Add  $\leftarrow$  csw  $\Rightarrow$  Aspect_Ratings["token"]["sw_count"]
13:      Detect  $\leftarrow$  sentiment_words  $\Rightarrow$  "polarity"
14:      if (polarity == positive) then
15:        psc = csw +  $v^+$  -  $v^-$ 
16:        Add  $\leftarrow$  psc  $\Rightarrow$  Aspect_Ratings["token"]["pscore"]
17:      else if (polarity == negative) then
18:        nsc = csw +  $v^+$  -  $v^-$ 
19:        Add  $\leftarrow$  nsc  $\Rightarrow$  Aspect_Ratings["token"]["nscore"]
20:      end if
21:    end if
22:  end for
23: end for
24: for each aspect  $\in$  Aspect_Ratings do
25:   ssc = (aspect["pscore"] - aspect["nscore"])/aspect["sw_count"]
26:   Add  $\leftarrow$  ssc  $\Rightarrow$  Aspect_Ratings["aspect"]["sscore"]
27: end for
28: return Aspect_Ratings

```

In general, the rating of various aspects related to a particular product can be performed by calculating the sentiment score "*SScore*", which is primarily based on the derivation of two leading indicators: the positivity score "*PScore*" and the negativity score "*NScore*", which can be obtained using the following equations.

$$PScore(a) = \begin{cases} \sum_{i=1}^k p_{r_i} + v_{r_i}^+ - v_{r_i}^- & \text{if } p_{r_i} > 0, \\ 0 & \text{if } p_{r_i} = 0, \end{cases} \quad (1)$$

where p_{r_i} refers to the number of positive words associated with the aspect "a" in the i th review, and $v_{r_i}^+, v_{r_i}^-$ refers to the number of positive and negative readers' votes, respectively.

$$NScore(a) = \begin{cases} \sum_{i=1}^k n_{r_i} + v_{r_i}^+ - v_{r_i}^- & \text{if } n_{r_i} > 0, \\ 0 & \text{if } n_{r_i} = 0, \end{cases} \quad (2)$$

where n_{r_i} refers to the number of negative words associated with the aspect “a” in the i th review, and $v_{r_i}^+, v_{r_i}^-$ refers to the number of positive and negative readers’ votes, respectively.

$$SScore(a) = \frac{PScore(a) - NScore(a)}{P_a + N_a}, \quad (3)$$

where P_a, N_a refers to the total number of positive and negative words, respectively, associated with the aspect “a” in all review data.

4.2. Extracting Preferences

Commonly, capturing the user preferences could be accomplished using two main methods; the first is the interactive method “conversational system” by using a web agent that makes a dialog with customers in the form of questions and answers and then extracts preferences from the user answering. The second method for extracting the user’s preferences, which is followed in this work, could be implemented by collecting all reviews made by the same user and then extracting features from the text “Preferences”. Therefore, the first step in this task is to extract aspect terms to identify the mentioned features by the user. The next is to compute the positivity and negativity scores as mentioned in Equations (1) and (2), respectively, with ignoring the readers’ votes, since the primary goal is to capture personal preferences only, where $v_{r_i}^+, v_{r_i}^-$ are considered equal to zero. Similarly, computing the sentiment score becomes straightforward according to Equation (3). Accordingly, the list of extracted aspects and their computed sentiment scores are entered into the recommendations engine in order to generate the appropriate list of candidate products.

4.3. Generating Recommendations

Given a list of rated products and a list of user preferences created by the first and second components in our system, respectively, the generation of recommendations becomes straightforward. Product ratings consist of aspects and the associated computed sentiment scores. Therefore, the first step is to search the rated products and find items that have aspects included in the user’s preferences list, generating a candidate products list. The following step is rating each product in the candidate list using the following equation, where the assigned weight is the sentiment score of the equivalent user preference “Personalized Rate” using the customer preferences exclusively.

$$Rate(t_l) = \sum_{i=1}^m a_i |w_i|, \quad (4)$$

where a_i refers to the i th aspect, and w_i refers to the associated weight with the given aspect of the l th item “t”.

Similarly, for rating different products in the candidate list, but this time including all mentioned aspects in each product regardless of the existence in the customer preferences “General Rate”, here, we have used equivalent weights equal to one. Finally, the candidate product list was sorted by the personalized and general rates, respectively, and the top items were recommended to the user.

5. Experiments and Results

In this work, we proposed integrating aspect-based sentiment analysis into a collaborative filtering-based recommender system to improve the system performance and precision of the predictions. Therefore, to validate the effectiveness of the proposed model, we conducted a series of experiments. We tested and evaluated it against several available and publicly accessed datasets. The following is a description of the datasets used in the experiment. Next is a review of the evaluation metrics and benchmarks used in our experiment—finally, a discussion of the achieved results is presented.

5.1. Datasets

We evaluate our proposed model on a dataset that contains ratings and review texts. The experiments were conducted using benchmark datasets researchers widely use to compare with baseline models collected from the Amazon website by Julian McAuley and Jure Leskovec (2013) [3]. The data span a period of 18 years, starting from June 1995 up to March 2013, including over 34 million reviews divided into 28 domains according to the item category, including cell phones, electronics, etc. We selected five categories for the experiment. Table 1 shows a summary of the selected data.

Table 1. A summary of the selected dataset.

Dataset	#Users	#Items	#Reviews
Automotive	133,254	47,539	188,387
Cell Phone and Accessories	68,039	7335	76,653
Electronics	811,032	80,613	1,189,549
Movies and TV	1,224,266	207,532	6,797,757
Video Games	228,516	20,335	364,484

5.2. Testing and Evaluation Metrics

5.2.1. Evaluation Metrics

We evaluated our model and compared it with the baseline models using mean-squared error (MSE). MSE is a metric used to measure the average of the squares of errors between the actual and predicted values; in our case, values are the ratings. Therefore, a lower value of MSE indicates a better performance of a model. Formally, given a test set of N reviews, where actual ratings are $r_1, \dots, r_n$, and the corresponding predicted ratings are $\hat{r}_1, \dots, \hat{r}_n$, the mean squared errors is defined by the following equation:

$$MSE = \frac{1}{N} \sum_{n=1}^N (r_n - \hat{r}_n)^2 \quad (5)$$

While our model generates ratings that consist of real numbers, actual user ratings consist of discrete values ranging from one to five. Therefore, converting the generated rates by our model into integer values is mandatory to evaluate our model properly. We have used the equal-width discretization approach, which separates all possible values into N number of bins, each having the same width; in order to accomplish this task, we used the following equation:

$$W = \frac{1}{N} (Max - Min), \quad (6)$$

where N is the number of bins or intervals, and Max and Min are the maximum and minimum ratings values, respectively.

5.2.2. Baseline Methods

For better evaluation of our model “SmartTips” in terms of rating prediction, a comparison of our model against multiple baselines was made, including the following models:

- **Matrix Factorization (MF)** [41]: Represents one of the most common collaborative filtering recommender systems, classified as a model-based approach. Typically, matrix factorization uses a rating matrix that characterizes both items and users as input for rating prediction by estimating two low-ranked matrices.
- **SVD++** [42]: This is one of the Latent factor-based models; SVD++ is an extension of the original Singular Value Decomposition model (SVD) that exploits the existence of both implicit and explicit feedback in order to improve the collaborative filtering-based recommendations performance.

- **ALFM-P** [9]: Aspect-based Latent Factor Model (ALFM) employs the item–property matrix made of word frequency to integrate ratings and review texts via the latent factor model.
- **REAO** [35]: Recommender system Exploiting Opinion mining (REAO) which consists of two major parts. The first is aspect-based opinion mining (ABOM) that employs a deep CNN model for aspect extraction with the help of the LDA technique to transform the extracted aspect terms into latent factors. The second is rating prediction, which computes the aspect-based ratings, forming a set of rating matrices that match a particular aspect.

5.2.3. Result Analysis and Discussion

Table 2 shows the performance analysis of our model with various baseline models. We report the mean squared errors for every selected domain of the dataset based on the authors' published results; we emphasized the best performance using the bold font; in the case of no results available related to the related domain, a cell was filled with a dash. Obviously, the SmartTips outperforms all baselines, except in the Video Game domain, where the ALFM-P model provides slightly better performance. In contrast, our model achieved an MSE of 0.515, and the ALFM-P achieved an MSE of 0.505. The average of the achieved MSE results is sufficient to justify our analysis. We can observe that the MF and SVD++ models provide a moderate performance compared with the ALFM-P and REAO models, which act better.

Additionally, hypothesis testing was conducted on the achieved results, proving that the predictions are feasible and statistically significant at $p < 0.05$. On the other hand, the proposed model guarantees to generate recommendations even if the user has no historical data, i.e., "New User", known as a *Cold Start Problem*, since our model examines the review text, which has sufficient information for capturing user preferences. Generally, the worst case is when the user provides a short review that includes a small number of aspect terms, or with an aspect-free review; our model will recommend highly rated products based on other users' feedback regardless of the current user preferences using the General Rate factor.

Table 2. Comparisons results of the proposed SmartTips model in terms of the rating predictions with the baseline models.

Dataset	MF	SVD++	ALFM-P	REAO	SmartTips
Automotive	1.715	1.439	0.565	0.663	0.473
Cell Phone and Accessories	2.323	2.112	0.364	-	0.330
Electronics	1.968	1.661	0.396	-	0.313
Movies and TV	0.627	0.544	0.412	-	0.393
Video Games	1.837	1.495	0.505	1.055	0.515
Average	1.694	1.450	0.448	0.859	0.405

6. Conclusions

Online customers' opinions represent a significant resource for customers and enterprises to extract much information that helps them make the right decision. Finding relevant data while searching the internet represents a big challenge for web users, known as the "Problem of Information Overload". Recommender systems have been recognized as a promising way of solving such problems. This work proposes a product recommendation system, "SmartTips", built on aspect-based sentiment analysis. The proposed model exploits customer feedback and applies the aspect term extraction model to rate various products. Several factors were considered, including readers' votes, aspect term frequency, opinion words frequency, etc. We examined the review text and meta-data as well. On the other hand, a similar scenario was proposed to capture user preferences for generating a recommendation for a specific user. Users' sentiments towards specific aspects are used to weigh aspects in candidate products to reflect the relevance of each

one and match user preferences appropriately. We conducted extensive experiments to ensure our model's implementation validity and verify its effectiveness, and the results show that the proposed model acts well. In order to assure its efficiency, we tested our model on the Amazon dataset that is widely used by researchers and compared it against the baseline methods. The results show that our model outperforms them in terms of the mean squared errors of the generated predictions. The proposed work has good prospects for prospective development, so we plan to incorporate the proposed recommendation model into a conversational-based recommender system in the future.

Author Contributions: Conceptualization, N.M.A.; methodology, N.M.A., A.A., A.M.A. and B.N.; software, N.M.A., A.A., A.M.A. and B.N.; validation, N.M.A., A.A., A.M.A. and B.N.; formal analysis, N.M.A., A.A. and A.M.A.; investigation, N.M.A., A.A., A.M.A. and B.N.; resources, N.M.A., A.A., A.M.A. and B.N.; data curation, N.M.A., A.A., A.M.A. and B.N.; writing—original draft preparation, N.M.A. and A.A.; writing—review and editing, N.M.A., A.M.A. and B.N.; visualization, N.M.A., A.A. and B.N.; supervision, B.N.; project administration, N.M.A., A.A., A.M.A. and B.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The researcher [Noaman M. Ali] is funded by a scholarship [EGY-6428/17] under the Joint (Executive Program between the Arab Republic of Egypt and the Russian Federation).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ali, N.M. Aspect-Oriented Analytics of Big Data. In *Proceedings of the 14th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2020)*; Matulevičius, R., Robal, T., Haav, H.M., Maigre, R., Petlenkov, E., Eds.; CEUR Workshop Proceedings: Tallinn, Estonia, 2020; Volume 2620, pp. 41–48.
2. Sulikowski, P.; Zdziebko, T.; Turzyński, D. Modeling Online User Product Interest for Recommender Systems and Ergonomics Studies. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4301. [\[CrossRef\]](#)
3. McAuley, J.; Leskovec, J. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, Hong Kong, China, 12–16 October 2013; pp. 165–172. [\[CrossRef\]](#)
4. Ali, N.M.; Novikov, B. A Multi-Source Big Data Framework for Capturing and Analyzing Customer Feedback. In *Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, Moscow and St. Petersburg, Russia, 26–29 January 2021; pp. 185–190. [\[CrossRef\]](#)
5. Hazrati, N.; Ricci, F. Recommender Systems Effect on the Evolution of Users' Choices Distribution. *Inf. Process. Manag.* **2022**, *59*, 102766. [\[CrossRef\]](#)
6. Ali, N.M.; Gadallah, A.M.; Hefny, H.A.; Novikov, B. An Integrated Framework for Web Data Preprocessing Towards Modeling User Behavior. In *Proceedings of the 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, Vladivostok, Russia, 6–9 October 2020; pp. 1–8. [\[CrossRef\]](#)
7. Al-Yazeed, N.M.A.; Gadallah, A.M.; Hefny, H.A. A Hybrid Recommendation Model for Web Navigation. In *Proceedings of the Seventh IEEE International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, 12–14 December 2015; pp. 552–560. [\[CrossRef\]](#)
8. Ali, N.M.; Gadallah, A.M.; Hefny, H.A.; Novikov, B.A. Online Web Navigation Assistant. *Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki* **2021**, *31*, 116–131. [\[CrossRef\]](#)
9. Qiu, L.; Gao, S.; Cheng, W.; Guo, J. Aspect-based Latent Factor Model by Integrating Ratings and Reviews for Recommender System. *Knowl.-Based Syst.* **2016**, *110*, 233–243. [\[CrossRef\]](#)
10. Zheng, Y.; Wang, D. A Survey of Recommender Systems with Multi-Objective Optimization. *Neurocomputing* **2022**, *474*, 141–53. [\[CrossRef\]](#)
11. Bag, S.; Kumar, S.K.; Tiwari, M.K. An Efficient Recommendation Generation using Relevant Jaccard Similarity. *Inf. Sci.* **2019**, *483*, 53–64. [\[CrossRef\]](#)
12. Jain, G.; Mahara, T. An Efficient Similarity Measure to Alleviate the Cold-Start Problem. In *Proceedings of the 2019 Fifteenth International Conference on Information Processing (ICINPRO)*, Bengaluru, India, 20–22 December 2019; pp. 1–8. [\[CrossRef\]](#)

13. Jain, G.; Mahara, T.; Tripathi, K.N. A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System. In *Soft Computing: Theories and Applications*; Pant, M., Sharma, T.K., Verma, O.P., Singla, R., Sikander, A., Eds.; Springer: Singapore, 2020; Volume 1053, pp. 343–352. [\[CrossRef\]](#)
14. Manochandar, S.; Punniyamoorthy, M. A New User Similarity Measure in A New Prediction Model for Collaborative Filtering. *Appl. Intell.* **2021**, *51*, 586–615. [\[CrossRef\]](#)
15. Tan, Z.; He, L. An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle. *IEEE Access* **2017**, *5*, 27211–27228. [\[CrossRef\]](#)
16. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R. A Content-based Recommender System for Computer Science Publications. *Knowl.-Based Syst.* **2018**, *157*, 1–9. [\[CrossRef\]](#)
17. Okura, S.; Tagami, Y.; Ono, S.; Tajima, A. Embedding-based News Recommendation for Millions of Users. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 1933–1942. [\[CrossRef\]](#)
18. Chen, R.; Hua, Q.; Chang, Y.S.; Wang, B.; Zhang, L.; Kong, X. A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access* **2018**, *6*, 64301–64320. [\[CrossRef\]](#)
19. Ullah, F.; Zhang, B.; Khan, R.U.; Chung, T.S.; Attique, M.; Khan, K.; Khediri, S.E.; Jan, S. Deep Edu: A Deep Neural Collaborative Filtering for Educational Services Recommendation. *IEEE Access* **2020**, *8*, 110915–110928. [\[CrossRef\]](#)
20. Xu, G.; Tang, Z.; Ma, C.; Liu, Y.; Daneshmand, M. A Collaborative Filtering Recommendation Algorithm Based on User Confidence and Time Context. *J. Electr. Comput. Eng.* **2019**, *2019*, 7070487. [\[CrossRef\]](#)
21. Agarwal, A.; Mishra, D.S.; Kolekar, S.V. Knowledge-based Recommendation System Using Semantic Web Rules Based on Learning styles for MOOCs. *Cogent Eng.* **2022**, *9*, 2022568. [\[CrossRef\]](#)
22. Khan, Z.A.; Raja, M.A.Z.; Chaudhary, N.I.; Mehmood, K.; He, Y. MISGD: Moving-Information-Based Stochastic Gradient Descent Paradigm for Personalized Fuzzy Recommender Systems. *Int. J. Fuzzy Syst.* **2022**, *24*, 686–712. [\[CrossRef\]](#)
23. Khan, Z.A.; Zubair, S.; Chaudhary, N.I.; Raja, M.A.Z.; Khan, F.A.; Dedovic, N. Design of Normalized Fractional SGD Computing Paradigm for Recommender Systems. *Neural Comput. Appl.* **2020**, *32*, 10245–10262. [\[CrossRef\]](#)
24. Jain, G.; Mishra, N.; Sharma, S. CRLRM: Category Based Recommendation Using Linear Regression Model. In Proceedings of the 2013 Third International Conference on Advances in Computing and Communications, Cochin, India, 29–31 August 2013; pp. 17–20. [\[CrossRef\]](#)
25. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Model. User-Adapted Interact.* **2002**, *12*, 331–370. [\[CrossRef\]](#)
26. Mu, R. A Survey of Recommender Systems Based on Deep Learning. *IEEE Access* **2018**, *6*, 69009–69022. [\[CrossRef\]](#)
27. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [\[CrossRef\]](#)
28. Zhang, L.; Luo, T.; Zhang, F.; Wu, Y. A Recommendation Model Based on Deep Neural Network. *IEEE Access* **2018**, *6*, 9454–9463. [\[CrossRef\]](#)
29. Batmaz, Z.; Yurekli, A.; Bilge, A.; Kaleli, C. A Review on Deep Learning for Recommender Systems: Challenges and Remedies. *Artif. Intell. Rev.* **2019**, *52*, 1–37. [\[CrossRef\]](#)
30. Chavare, S.R.; Awati, C.J.; Shirgave, S.K. Smart Recommender System using Deep Learning. In Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 590–594. [\[CrossRef\]](#)
31. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10. [\[CrossRef\]](#)
32. Covington, P.; Adams, J.; Sargin, E. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198. [\[CrossRef\]](#)
33. Ni, J.; Li, J.; McAuley, J. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP) and the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; ACL: Stroudsburg, PA, USA, 2019; pp. 188–197. [\[CrossRef\]](#)
34. Baizal, Z.K.A.; Widyanthoro, D.H.; Maulidevi, N.U. Computational Model for Generating Interactions in Conversational Recommender System Based on Product Functional Requirements. *Data Knowl. Eng.* **2020**, *128*, 101813. [\[CrossRef\]](#)
35. Da’u, A.; Salim, N.; Rabi, I.; Osman, A. Recommendation System Exploiting Aspect-based Opinion Mining with Deep Learning Method. *Inf. Sci.* **2020**, *512*, 1279–1292. [\[CrossRef\]](#)
36. Da’u, A.; Salim, N.; Rabi, I.; Osman, A. Weighted Aspect-based Opinion Mining Using Deep Learning for Recommender System. *Expert Syst. Appl.* **2020**, *140*, 112871. [\[CrossRef\]](#)
37. Ghasemi, N.; Momtazi, S. Neural Text Similarity of User Reviews for Improving Collaborative Filtering Recommender Systems. *Electron. Commer. Res. Appl.* **2021**, *45*, 101019. [\[CrossRef\]](#)
38. Asani, E.; Vahdat-Nejad, H.; Sadri, J. Restaurant Recommender System Based on Sentiment Analysis. *Mach. Learn. Appl.* **2021**, *6*, 100114. [\[CrossRef\]](#)

-
39. Ray, B.; Garain, A.; Sarkar, R. An Ensemble-based Hotel Recommender System Using Sentiment Analysis and Aspect Categorization of Hotel Reviews. *Appl. Soft Comput.* **2021**, *98*, 106935. [[CrossRef](#)]
 40. Ali, N.M.; Alshahrani, A.; Alghamdi, A.M.; Novikov, B. Extracting Prominent Aspects of Online Customer Reviews: A Data-Driven Approach to Big Data Analytics. *Electronics* **2022**, *11*, 2042. [[CrossRef](#)]
 41. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [[CrossRef](#)]
 42. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434. [[CrossRef](#)]