applied sciences

MDPI

Article

Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities

Zeshan Fayyaz [1], Mahsa Ebrahimian [1], Dina Nawara [1], Ahmed Ibrahim [2] and Rasha Kashef [1,*]

[1] Electrical, Computer and Biomedical Engineering Department, Ryerson University, Toronto, ON M5B 2K3, Canada; zeshan.fayyaz@ryerson.ca (Z.F.); mahsa.ebrahimian@ryerson.ca (M.E.); dina.nawara@ryerson.ca (D.N.)

[2] Computer Science Department, Wilfried Laurier University, Waterloo, ON N2L 3C5, Canada; a24ibrah@uwaterloo.ca

* Correspondence: rkashef@ryerson.ca

Received: 9 October 2020; Accepted: 30 October 2020; Published: 2 November 2020

check for updates

Abstract: Recommender systems are widely used to provide users with recommendations based on their preferences. With the ever-growing volume of information online, recommender systems have been a useful tool to overcome information overload. The utilization of recommender systems cannot be overstated, given its potential influence to ameliorate many over-choice challenges. There are many types of recommendation systems with different methodologies and concepts. Various applications have adopted recommendation systems, including e-commerce, healthcare, transportation, agriculture, and media. This paper provides the current landscape of recommender systems research and identifies directions in the field in various applications. This article provides an overview of the current state of the art in recommendation systems, their types, challenges, limitations, and business adoptions. To assess the quality of a recommendation system, qualitative evaluation metrics are discussed in the paper.

Keywords: recommendation systems; machine learning; validation; business applications

1. Introduction

The internet and modern web services have been increasing within the last few decades; a surplus of information is now accessible to everyone [1]. It can be challenging for users to filter through all this information and take away essential aspects. Many online e-commerce firms recommend products to their users, selling millions of products on one platform. For an everyday user, browsing through all possibilities can be overwhelming; this can cause information overload. Recommender systems aim to solve that information overload problem while personalizing the user experience by delivering accurate, personalized recommendations of items/products to users according to their preferences [2]. A recommendation system (RS) aims to predict if an item would be useful to a user based on given information [3]. The use of these systems has been steadily growing within the last few years, where they are used in retail and e-commerce firms like eBay and Amazon. These companies acquire massive users' data and tailor the RSs to meet the users' and business needs [4,5]. RSs are widely utilized in e-commerce and retail; they are also used in many other industries such as healthcare, transportation, and agriculture [6–8]. High-quality RSs positively impact the users' experience and the overall enterprises' revenue or decision. RSs have attracted many researchers for the past years, and various literature reviews were conducted, addressing different RSs' features, algorithms, and challenges [1,9–14]. Yet, none of these reviews touched on all RSs' aspects holistically. In [9], authors have focused on categorizing RSs based on data they have used. A survey of RSs using only social networks is provided in [10] and location-based RSs using social networks [11]. RSs are surveyed in [12] from their application point of view. A review in [13] has focused on algorithms of RSs [13] while the characteristics of RSs are summarized [14].

In this paper, we present a comprehensive guide to RSs. We highlight different RSs categories and analyze the issues associated with current RSs, including cold start, data-sparsity, scalability, and diversity. Also, we demonstrate how to evaluate the RSs' performance using different metrics such as recall, precision, accuracy, ROC (Receiver operating characteristic) curve, and F-measures. Moreover, we outline the adoption of RSs in many industries and domains. This paper is structured as follows: Section 2 presents the recommendation categories, Section 3 highlights the main challenges in RSs, Section 4 explains the different evaluation metrics, Section 5 introduces the business adoptions of RSs, and Section 6 is for the conclusion and future directions.

2. Recommendation System Categories

RSs [15] are categorized into collaborative filtering, content-based, utility-based, demographic-based, knowledge-based, and hybrid-based. The most commonly used filtering approaches use content-based and collaborative filtering. In this section, a brief explanation of these categories is provided.

2.1. *Collaborative-Filtering Recommendation Systems*

Collaborative Filtering evaluates products using users' ratings (explicit or implicit) from historical data [16]. It works by developing a database of the user's preferences for items. Active users will be mapped against this database to reveal the active user's neighbors with similar purchase preferences. Collaborative filtering techniques are classified into item-based filtering and user-based filtering [17]. User-based techniques go through two main stages to forecast items' ratings for a specific user. The first stage locates similar users to the target user. The second stage obtains rates from similar users to the active user, then using them to produce recommendations. There have been many collaborative filtering algorithm measures that calculate the similarities among users. The commonly used similarity measures in the literature include mean-squared difference, Pearson correlation, cosine similarity, Spearman correlation, and adjusted cosine similarity [18,19]. Collaborative filtering is the commonly used choice for RSs, and it does not require domain knowledge because the embeddings are automatically learned. Embedding of items in a recommender system refers to mapping items to a sequence of numbers. This way of representing items with learned vectors, is used to train algorithms to find the relationship between items and extract their features. Next, an advantage of collaborative filtering is that it generates models that help users discover new interests. Finally, collaborative filtering is a great starting point for other RSs, as the RS only requires the rating matrix R to develop a factorization model. The rating matrix R is a two-dimensional matrix of n users and m items; each entry in this matrix, rij represents the rating provided of user i to item j. Although being favorable in many aspects, collaborative filtering has several disadvantages, such as the cold-start problem, which we will elaborate on in Section 3.1 of this paper.

2.2. *Content-Based Recommendation Systems*

Content-based approaches attempt to build a user profile to predict ratings on unseen items. Successful content-based methods utilize tags and keywords. Measuring the utility of content-based filtering is commonly calculated by using heuristic functions, such as the cosine similarity metric. Content-based filtering can be employed in many cases, where the features' values can easily be extracted. Content-based filtering is not typically used in cases where features values must be manually entered. This can be manageable for small datasets, but when thousands of new products are being added daily, this task is impossible. Content-based filtering does not require other users' data, as the predicted recommendations are user-specific. Thus, these techniques scale up the system to handle many users. Content-based filtering is user-independent since this system only requires analyzing the items and user profile for recommendations. Opposite of collaborative filtering, content-based filtering does not experience cold-start issues. New items or products are suggested before a substantial list of users assigns a rating. Content-based filtering has several drawbacks. Firstly, if no enough information is provided in the content to differentiate products precisely, the recommendation will not be accurate. These techniques require intensive domain knowledge. Secondly, content-based systems offer a limited degree of novelty since they must match up the features of profiles and items [18,19].

2.3. *Demographic-Based Recommendation Systems*

As various quantitative research papers have displayed, collaborative filtering techniques can be enhanced by demographic correlation [20]. Demographic RSs can generate recommendations by categorizing users based on demographic attributes. Demographic RSs are especially useful when the amount of product information is limited. Demographic RSs aim to tackle and solve the scalability and cold-start problems. This system employs user attributes as demographic data to obtain recommendations (i.e., recommend products based on age, gender, language, etc.) [21]. The key advantage of demographic filtering RSs is that they are fast and straightforward in obtaining results using a few observations. These approaches also do not acquire the user ratings that are essential in content-based and collaborative-based filtering techniques. Demographic-based filtering techniques have several disadvantages. For example, the entire information collection for users is impractical, considering the security and privacy issues involved. Secondly, demographic filtering is mainly based on user interests, which forces the system to recommend the same item to users of related demographic profiles. Another challenge is the difficulty of modifying a customer profile when preferences change; this is known as the stability vs. plasticity problem.

2.4. *Utility-Based Recommendation Systems*

Utility-based RS provides recommendations based on generating a utility model of each item for the user. This system builds multi-attribute users' utility functions and recommends the highest utility item based on each item's calculated user-utility explicitly [22]. Utility-based RSs are useful because they can factor non-product attributes into utility functions, such as product availability and vendor reliability. They generate utility computation, which allows them to check both real-time inventory and features of an item. It enables the visualization of its status to the user. Utility-based systems do not hold on to long-term generalizations about their users. Instead, they evaluate a recommendation based on the user's current needs and the available options. A disadvantage of the utility-based system occurs when the products are not descriptive enough. They do not contain enough listed utility features; that could hide a recommendation to a user even if it fits that particular user's preferences [23].

2.5. *Knowledge-Based Recommendation Systems*

Knowledge-based RS uses explicit knowledge about products and users to create a knowledge-based criterion to generate recommendations [23]. A knowledge-based RS does not require an initial large amount of data, as its recommendations are independent of the user's ratings [24]. It recommends items based on the user's preferences by evaluating the products that meet the user's needs. Knowledge-based RSs are noted to be advantageous for several purposes. For example, they can avoid the typical ramp-up problem associated with machine learning approaches to recommendations. Typically, exemplary systems cannot learn until the user has rated many items. Knowledge-based RSs avoid this issue since their recommendations are not dependent on a base of user ratings. They also do not need to gather information about a particular user because the recommendations are independent of the user's tastes too. Due to these factors, knowledge-based systems are valuable as stand-alone systems, and they are also considered complementary to other types of RSs. One major disadvantage of knowledge-based RSs is the potential knowledge acquisition bottleneck caused by explicitly defining recommendation knowledge. Knowledge acquisition is the process of constructing the rules and requirements needed for a knowledge-based system, and it is done by gaining knowledge via rules, objects, and frame-based ontologies. Batesonian theories were used to guide the process of further learning of knowledge acquisition [24].

2.6. Hybrid-Based Recommendation Systems

Hybrid systems are combining two or more techniques to obtain better performance. Their main target is to eliminate the drawbacks of the individual ones. Some of the combination strategies are discussed next. Figure 1 shows various hybrid strategies.

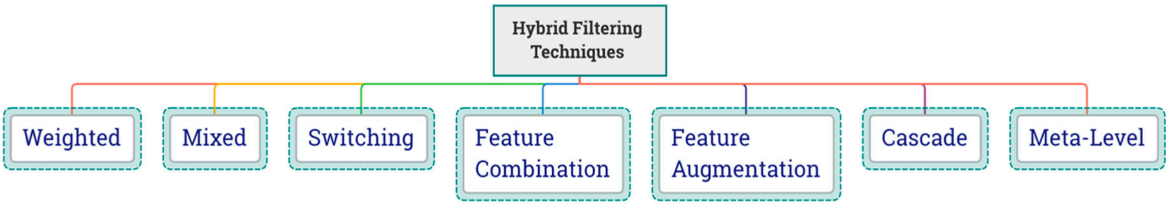

Figure 1. Hybrid filtering strategies.

2.6.1. Weighted

Weighted-hybrid recommender aggregates the results of all combined recommendation approaches, then calculates the recommended item/value's score. A linear combination of multiple recommendation scores method is used. Systems initially give all recommenders an equal weight, then progressively adjust the weighting as predictions of user ratings are verified or not. However, this model implicitly assumes that individual techniques' relative value is uniform among possible items, which is not always true [15,23].

2.6.2. Switching

A switching method selects one recommender from the constituents. For a different user/profile, another system may be chosen. For example, if the content-based technique cannot make an accurate recommendation with high confidence, then another method like the collaborative procedure is attempted. This method does not avert all drawbacks experienced by RSs (i.e., the ramp-up problem). This hybridization method assumes that there is a reliable criterion for which to make the switching decision. Once the switching decision is made, the other unchosen components do not have a role in the left recommendation process [15,16,23].

2.6.3. Mixed

A mixed hybrid method is practical when many recommendations are needed simultaneously. The mixed hybrid method shows recommendations of its components side-by-side in a consolidated list. This hybridization method does not try to consolidate evidence between recommenders. Combining several independent lists is a challenging process for this method. Standard techniques cover either merging based on predicted rating or based on the recommender's confidence [15,23].

2.6.4. Feature Combination

Feature combination allows the combination of one technique's complementary features, for example, a collaborative-based recommendation, into an algorithm planned to process data with another method (for example, content-based recommendation). Content-collaborative merger is achieved by dealing with collaborative informational as an additional feature data linked to each model and utilize content-based techniques over this built up dataset. This technique lets the system consider collective data without totally relying on it so that the system's sensitivity to the number of users who rated an item is reduced [16].

2.6.5. Cascade

Cascade method is an organized process used to form a strictly hierarchical hybrid, such that a weak technique with low priority cannot cancel the decisions made by a higher priority or stronger one, but rather can improve them. The lower priority recommender is utilized in breaking ties in the scoring of the stronger and higher priority ones. The lower priority technique is not used on the already well-differentiated items by the first one. Also, it is not used on the poorly rated items, so they will not be recommended. The cascading method is resilient to the noise in the low priority technique's operation, as ratings can only be improved, not reversed [15–17].

2.6.6. Feature Augmentation

This method is applied to generate an item's rating, then integrate this information into the processing of the next recommendation technique. A new feature for every item is generated by feature augmentation, using the contributing domain's recommendation logic. Feature augmentation is used when there is a well-developed main recommendation component, and there is a need to add additional knowledge features or sources. Unlike the cascade model, in the augmentation hybrid method, the output features of the first recommender are included in the features used by the second one [15,23].

2.6.7. Meta-Level

The meta-level hybrid uses an output model, which is learned by a recommender to be used as another one's input. This method is not similar to feature augmentation. Feature augmentation hybrid uses a learned model's general features as input for a second one, whereas in a metalevel hybrid, the entire learned model is used as an input. The recommender is not functioning with raw profile data. Deriving a meta-level hybrid from any provided pair of recommenders is not always an easy task. Since the contributing recommender must generate a model used as input by the actual recommender; however, not all recommendation techniques can achieve this. A benefit of this technique is that the learned model indicates a compressed representation of the user's preference. A collaborative approach can operate on this compact representation easier than working on raw rating data [15–17].

3. Challenges in Recommendation Systems

It is challenging to measure RSs' performance due to the organization's changing demands using and deploying it. Generally, the most indicative measure is user satisfaction. Even though it is not possible to compute users' satisfaction by using a heuristic formula, we can still measure the performance of RSs based on how well they can handle common issues. In this section of the review paper, we provide an understanding of the metrics used to measure the performance of RSs against main challenges, including cold-start, accuracy, data sparsity, scalability, and diversity.

3.1. Cold-Start

The term 'cold start' stems from automobiles. When the engine is cold, they have difficulty starting up, but they have no problems running once they reach their optimal temperature. The same problem can be applied to RSs. When there is insufficient information or metadata available, a RS does not perform optimally. Cold starts can be classified into two distinct subsets: product cold starts and user cold starts [25]. Whenever a new product is displayed on an e-commerce site, it goes through the product cold start, and this means that there are no reviews due to the lack of user interaction. If there are not enough user interactions, the RS will not know when to display the ad related to that product. The cold-start behavior occurs when a user creates an account for the first time and does not have any product preferences or history available to base recommendations. The cold start problem always exists for new or existing users. For example, Tom searches for new televisions on an e-commerce site; within a week, he purchases one and is no longer interested in purchasing televisions; what should the RS display now? Users will always be interested in new and different things. Analyzing the metrics and methods for cold-start recommendations, we find that the Bayes classifier is most used [26]. Bayesian models are graphical models used in probability and artificial intelligence. In model-based RSs, a form of Bayesian reasoning is likely to be applied, whether it is content- or collaborative-based. The most popular method of utilizing Bayesian models is the naive Bayes model [25]. Despite its simplicity, it has proved to be the most accurate. In the naive Bayes classification, different attributes are assumed to be mutually independent features of the items [26]. With this, one can estimate new item's characteristics with a set of attributes not found in the training data. The projection in WALS (weighted alternating least squares) and heuristics are used to address the cold-start problem to a certain degree. For the projection in the WALS method, if there is a new item not seen in training, yet the system has a few interactions with users, the user's embeddings for this item can be calculated easily, without the need to retrain the entire model as shown in Equation (1).

$$min_{u_{i_0} \in R^d} \parallel A_{i0} - u_{i0} V^T \parallel \tag{1}$$

Equation (1) is equivalent to one iteration in the WALS method, where the user's embeddings are kept exact, and the system solves for the embedding of the new item, and the same process can be performed for a new user to keep the model up to date. In the heuristics methods that generate fresh items' embeddings, the embeddings can be approximated if the system does not have interactions. This is completed by taking the average of the item's embeddings of the same category.

3.2. Data Sparsity

Data sparsity results from the fact that the users only intend to rank limited items. Most RSs group the ratings of similar users; however, the reported user–item matrix has empty or unknown ratings (up to 99%) because of the lack of incentives or user knowledge to rate items [27]. Therefore, RSs can provide unreasonable recommendations to those who provide no feedback or ratings. For example, suppose we assume that an online bookstore sells two million different books, with X number of users (active or cold). In that case, each consumer is exemplified by an integer feature matrix, with 2 million elements, and the value of each component corresponds to the rating given by the consumer to a specific book. This matrix is called the consumer-product interaction matrix [28]. In most large-scale applications, both the numbers of consumers and products are enormous. Therefore, the majority (up to 99%, on average) of these matrix elements are 0. Comparing any two users for a specific item, it is very probable that both elements are 0, resulting in a sparse matrix [29]. Many techniques aim to mitigate the data sparsity issue by modeling users' preferences from their behaviors and trusted social connections. Trust has been extensively used to achieve significant benefits to the robustness of RSs [30]. Trust is described as the belief towards others' ability to provide accurate ratings (explicit and implicit). Many argue that it is possible to calculate trustworthiness based on the trust chart encoded by Epinions.com (a website where users can review items). The trust value can be calculated by measuring users' distance in the number of arcs connecting those users [31]. This offers a trust-aware RS that depends on a web of trust for defining how a user can trust another user. A trust network is constructed by aggregating every trust statement. A trust network consists of users and trust statements, represented by nodes and directed edges, respectively. These methods have significantly lowered the mean error of predictive accuracy. Many trust-based approaches have been introduced with significance given to the merge approach [32]. The merge incorporates the active users' trusted neighbors, seeking to enhance the overall predictive accuracy of RSs. Specifically, the ratings of a trusted neighbor of an active user are merged by averaging on frequently rated items as per the similarity between the active user and trusted neighbor.

3.3. Scalability

Scalability problems have been significantly raised due to the fast growth of e-commerce sites. Modern RS methodologies are required to generate quick results for large scale applications. RSs can search for many potential neighbors in real-time, but the demands of modern e-commerce sites require them to search for a larger number of neighbors. Algorithms also experience performance issues for consumers with large amounts of information [32]. For example, if a site has tens of thousands of data points for one user, it can be difficult and tedious to find a relevant neighbor for a given neighbor. Filtering algorithms that utilize nearest-neighbor techniques need an increase in computation power due to the massive increase in products or users. For a platform that has millions of users and products, scalability is a serious issue. A common technique to reduce scalability issues is by using one-dimensionality reduction [33,34]. Clustering techniques can be utilized to mitigate scalability issues. Their primary function is to segment the users using a clustering algorithm and use each segment as a neighborhood. Next, for any active user, its neighborhood is selected by looking into the partition, and the partition is used as the user's neighborhood. Upon completing the neighborhood selection, classical filtering algorithms can be implemented to generate a prediction [35]. There are two significant benefits of implementing clustering techniques. Firstly, it alleviates the sparsity of the data set. Secondly, it divides the data into smaller partitions, which significantly reduces prediction generation speeds. Singular value decomposition (SVD) has also been used to reduce the scalability issue [34]. SVD is used for dimensionality reduction. SVD produces a set of uncorrelated eigenvectors. Customers and products are each represented by a unique eigenvector. This process allows customers who have rated similar (but not the same) products to be mapped by the same eigenvectors. Once the $\mathrm{n} \times \mathrm{m}$ rating matrix is decomposed into SVD component matrices, predictions can be generated by calculating the cosine similarities (dot product) between n-pseudo customers and n-pseudo products.

3.4. Diversity

In various situations, recommendation systems may provide suggestions of either similar items or more diverse ones. Simultaneously, the most accurate results are obtained by recommending items/objects based on user or objects' similarity. This is known as the diversity issue, where recommendations are based on overlapping instead of differences. This exposes the user to a narrower selection of objects, while highly related niche items may be overlooked. The diversity of recommendations allows users to discover objects which they would not readily find for themselves. One apparent concern is that if an algorithm focuses strictly on enhancing diversity, accuracy would be lost [36]. The diversity of a RS can be evaluated by two measures; surprisal and personalization [36]. Self-information or 'surprisal' measures are used to gauge the RS's ability to generate unpredictable results, which measures the unexpectedness of an item/object proportional to its global popularity. Personalization is the uniqueness of different user's recommendation lists, known as inter-user diversity, and the inter-list distance can easily calculate this. The accuracy threshold must be preserved to address diversity issues while maintaining item recommendations [37]. Cases in which the RS is overly focused on accuracy is known as overconcentration. The LCM (linear time closed itemset miner) can increase diversity by finding efficient frequent item-sets [37].

3.5. Habituation Effect

Recommendation interfaces are considered as a critical element of marketing strategies and it can be considered as a means of distributing the marketing content. In order to optimize the performance of the interface, a number of elements can be explored, such as the number of recommendations, images of the recommended item, item descriptions, and layouts [38,39]. As customers are immersed with massive information, especially marketing content, the habituation effect usually appears, which ends in the banner blindness phenomenon. Thus, even recommendations that are optimal from the algorithmic perspective, they may provide inaccurate results unless they are visualized to the user in a better way. To avoid the banner blindness phenomenon, marketers usually use techniques based on increasing visual intensity [40] of presented objects with the use of animations and flickering effects [41]. The habituation effect can best be reduced with multi-criteria decision analysis (MCDA) of features of recommending interfaces taking into account their visual intensity, attention represented by fixations measured with eye-tracking and time required to attract attention after a website is loaded.

4. Evaluation Metrics

Evaluating a RS requires identifying the characteristics that make an excellent RS and identifying how they should be quantified. The typical metrics for assessing a RS's performance are recall, precision, accuracy, ROC curves, and F-measure.

4.1. Recall and Precision

Information retrieval (IR) is focused on retrieving relevant documents from a pool similar to RSs' function of recommending interesting and applicable items from a pool of resources. The IR field is considered an adequate provider of tools for RSs, such as measurement metrics. Two key metrics are 'precision and recall'. Precision indicates the fraction of relevant items among all the recommended items to a user, and recall represents the number of relevant recommended items to the total number of items that should be recommended. A relevant item is the one that the user finds appealing. Precision and recall metrics are calculated by computing a confusion matrix, similar to the one in Table 1. The confusion matrix represents the four possible outcomes of any recommendation, and if the recommended item is relevant to a user, it will be considered successful, otherwise it is not successful.

Table 1. Confusion matrix for a recommender system.

	Successful Recommendation	**Not a Successful Recommendation**
Recommended	a	b
Not Recommended	c	d

Where "a" represents the number of true positive items that are originally recommended and successfully retrieved for recommendations by the RS. The value of "b" indicates the number of items not successfully suggested by the RS, although they are labeled as recommended. The value "c" represents the number of disqualified items that are recommended by the RS. The true negative values, "d", refers the number of items that are labelled and retrieved as 'not recommended'.

A good RS tries to optimize both metrics simultaneously. For instance, it can recommend many products/items to the user, obtaining maximum coverage. The *Precision* would still be as low as the ratio of useful products/items in the pool.

$$Precision = \frac{a}{a+b} \tag{2}$$

$$Recall = \frac{a}{a+c} \tag{3}$$

4.2. Accuracy

The decision to select a RS based on an evaluation metric is complex and depends on the organization's needs. Generally, predicted ratings are used as an evaluation metric for a system. Deciding the accuracy of RS is not straight forward due to the lack of an explicit method in determining whether a recommendation is precise or not [42]. To assess an RS's accuracy, one must search for low prediction errors by using split-validation of data for offline comparisons. For example, suppose we submit 80% of a user purchase history dataset to a RS and ask to predict the rest. In that case, we can calculate the system's accuracy based on true recommendations, as shown in Equation (4).

$$Accuracy = \frac{Number\ of\ successful\ recomendations}{Total\ number\ of\ recommendations} \tag{4}$$

Particularizing the metric to RS, accuracy is used for evaluation in many cases; for example, in scoring an algorithm, the root of the mean square error (RMSE) is used [43]. Other alternatives of the RMSE are the mean average error, the normalized mean average error, and mean square error. RMSE is most appropriate since it measures all ratings' inaccuracies, whether they are positive or negative. RMSE is recommended to be used in cases where the errors cannot be differentiated. For example, predicting a rating difference between 1 to 2 stars may not be as important as a rating difference between 2 to 3 stars.

4.3. ROC Curve

Receiver operating characteristic (ROC) analysis is substitutional to precision/recall. A precision versus recall curve is shown in Figure 2. The higher the precision is, the lower the recall values

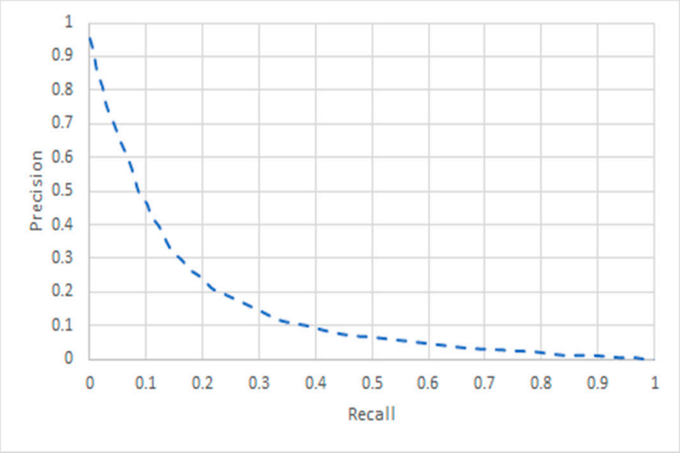

Figure 2. Precision–recall curve.

A ROC curve indicates fallout versus recall, as shown in Figure 3. ROC analysis is aimed to retrieve the relevant items without retrieving the irrelevant ones [44]. This is obtained by having the recall maximized, which is also called the (true positive rate) while having the fallout minimum, which is called (false positive rate). ROC curves are utilized to visually describe the trade-off among recall and precision when the threshold is changed, which helps us classify an item as "to be recommended" and "not to be recommended" [45]. The optimization of the ROC, precision, and recall curves are similar. In Figure 4, optimizing the recall and precision values can be achieved by pushing the curves' peak towards the point Precision = 1 and Recall = 1. An ideal predictive system produces a ROC curve that ultimately reaches all of the relevant items encountered and then headed for the remaining items. ROC curves assume binary relevance, same as precision and recall measures. Items are either classified to be a successful recommendation or an unsuccessful recommendation. However, by considering this, the order among relevant items does not impact the ROC metric. If all relevant items show before the non-relevant items, an ideal ROC curve is obtained [16]. To utilize the ROC curve as a measurement of performance, we can analyze the area underneath the curve [46]. The area under the ROC curve is the probability of the system's ability to correctly select between two items, such that one item is randomly picked from the set of acceptable items, and the second item is chosen from the set of unsatisfactory items.

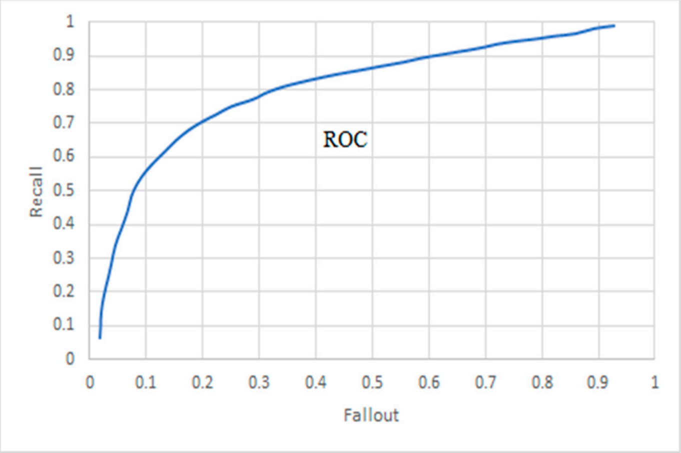

Figure 3. ROC (Receiver operating characteristic) curve for the recall against fallout.

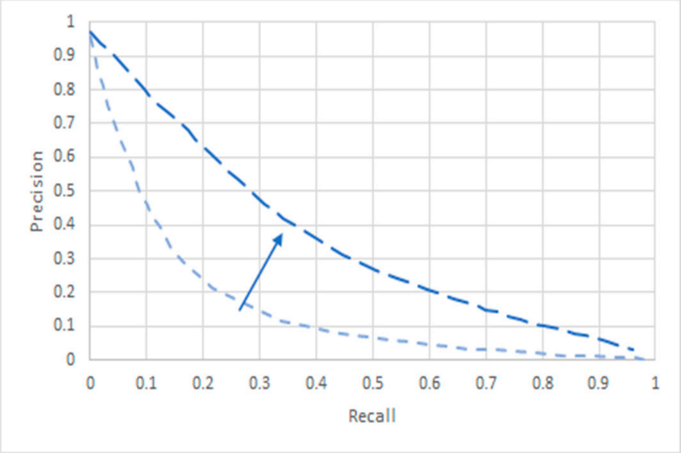

Figure 4. Simultaneous optimization of the precision and recall.

4.4. F-Measure

F-measure is another measure derived from precision and recall, and it exhibits the behavior of both recall and precision metrics. F-measures can be a more useful metric than precision and recall; since precision and recall give various information that can complete one another when combined. F-measure will reflect the metric that excels more than the other one. From a probability perspective, F-measure is the number of tests that need to be implemented to detect the first failure [47]. By varying β, the value of F_β gives more weight to one metric over the other. Yet, the most common F-measure is known as the consistent mean of precision and recall (F1), where $\beta = \frac{1}{2}$. Note that the maximum value F-measure can be is 1, which means that all predictions are accurate recommendations. It is useful to understand ranked retrieval. Here, Precision@K is the ratio of the top-k relevant items, and Recall@K is the ratio of relevant items in top-k. Ranked retrieval works under the assumption that the user will examine only the top-k results. Knowing this, [48] shows that as the value of *k* increases, F-measure will also increase because of the recall value's upward nature. Equations (2) and (3) show the calculations of both the F_β and F1.

$$F_\beta = \frac{precision \ \times recall}{(1-\beta)\times precision + \ \beta \times recall} \tag{5}$$

$$F_1 = \frac{2\times precision \ \times recall}{precision + \ recall} \tag{6}$$

5. Business Adoption and Applications

RSs were once a novelty technique used by very few e-commerce sites. Now, they have transformed into a serious tool that is drastically shaping the e-commerce world. Many of the largest e-commerce businesses utilize RSs to help users determine what they want, alleviating the information-overload problem. However, RSs are not limited to marketing products; they have been widely developed in the service industry. They can provide recommendations in many different areas ranging from location-based information to movies, music, images, books, etc. Location-based data helps users find their path or predict their next location to save time and cost [49]. Moreover, various parties can benefit from recommendations in their decision-making process, such as farmers [50], healthcare workers [51], or tourists [52]. For instance, farmers who seek optimal production plans to avoid loss can benefit from the agricultural-items predictions of RSs designed to recommend the best agricultural items (crops) cultivation options to farmers [50]. In the healthcare industry, RSs play a remarkable role in the decision-making process. The studies showed that health RSs (HRS) had been used for dietary, activity assistance, and educational purposes [51]. Therefore, in this section, we classify RSs based on their business adoption into five categories of e-commerce, transportation, agriculture, healthcare, and media.

5.1. Recommendation Systems in e-Commerce

RSs are aimed at providing customized recommendations of products to customers of websites. They learn from the customer and recommend relevant products to the user. These systems personalize the experience while attaining user interest. Simply, there are three ways a product is recommended based on (i) top overall sellers on a site, (ii) the demographics of the customer, or (iii) the previous purchase history of the customer. Implementing a RS personalizes a site as they adapt and are unique to each customer. On the other hand, RSs can enhance sales of e-commerce sites in three ways.

- *Browsers into buyers*: Visitors to an e-commerce site often look over products without buying anything, but if a site displays relevant recommendations to a user, they are more likely to purchase.
- *Cross-sell*: Recommendation techniques suggest additional products to the users, apart from the one they are already buying. With this, the average order size should increase over time
- *Loyalty:* In an era where a competitor's site can be visited by a mere click or two, loyalty is essential. RSs personalize the site for each user, which builds the user-site relationship. The more a customer uses a system, the more they are training the system, the more loyal a customer becomes, which also improves the quality of recommendations, over time.

RSs are also used as a technology to enable businesses to target customers and make offers to them. For instance, search engines and advertising companies rely on showing effective suggestions to users based on their behavior. Different recommendation methods have been used to achieve these goals, such as statistical methods, raw retrieval, attribute-based methods, user-to-user, or item-to-item correlation techniques. Each process may require different input types such as customer purchase history or information from users' communities, such as their ratings to items [4]. These methods usually result in providing prediction, suggestion, or ratings of items. Some applications of RSs in e-commerce businesses based on [4] are on Amazon.com, Drugstore.com, CDNOW (Compact Discs Now), eBay, Reel.com, and MovieFinder.com. Amazon.com has been noticed by more researchers [5] because of its highly personalized website and its launch of item-based collaborative filtering, which is one of the commonly used types of RSs. YouTube and Netflix have used collaborative filtering for video and movie recommendations. There are also some criticisms of the collaborative filtering method as it can be poorly be justified because it solely depends on rating data and disregarding the content data [53]. With the increasing number of RSs in e-commerce, companies need to select the best recommendation algorithm; therefore, Geuens et al. [54] has proposed a framework to choose an optimal collaborative filtering algorithm. They have used K-nearest Neighbour (KNN) as a classification method for this purpose and tested the result on two real datasets of women's clothing and furniture. A recent study [55] provides a user interface for an e-commerce website based on users' behavior using a deep neural network method. Their analysis revealed the effect of a website layout to recommended items based on the user's behavior. Customers reviews are used widely in RSs; in [56], A recent survey focuses on sentiment analysis on text reviews. They have illustrated that, generally, there are three types of RSs using text reviews based on words, topics, and opinions. Even though RSs have been utilized in e-commerce for a long time, there are still challenges in this area. For example, there is the issue of scalability and data latency. A RS on a website with a large number of users should respond in real-time. Also, there is a data sparsity issue of rating datasets as not all the customers will rate all the products.

5.2. Recommendation Systems in Transportation

RSs can assist in diverse ways with the increasing use of Global Positioning System (GPS)-enabled devices, especially mobile devices. Because information overload problems become worse when using mobile devices. The development of wireless communication services and position detection techniques such as RFID or GPS have promoted location-based information systems. RSs play a significant role in path recommendation, smart transport application of goods [8,57], tourism industry [58–60], or venue recommendation [61]. To predict users' location and suggest the best pathways, RS can use users' location data and integrate it with public transportation system data. An algorithm is proposed in [49] to predict users' path and recommend the best bus line. A prototype of their proposed algorithm is also developed for Android cell phones. Inputs of this system are the users' route and bus lines' data. RSs are used in smart transport applications with different methods, such as optimization [57] or clustering [8]. For instance, a clustering-based RS is developed in [8] for the transportation of goods. They reduced data dimensionality using principal component analysis (PCA) and applied a K-means method for clustering transporters based on their distance to users. This system generates a recommendation of the best available transporters for a user based on their location and transporters' profile. Some RSs aim to help users, mainly tourists, to find places such as the best restaurants. The proposed RS in [59] takes users' personal information and restaurants' attributes as input and applies a Bayesian network to calculate recommendation score and show recommended restaurants. Another restaurant RS is proposed in [62]. Collaborative filtering recommendation is used in [60] to help users explore the attractions of a city. They used histories of user's activities and visitors log to recommend locations. The proposed system enables users to collaboratively share their photos and experiences and provide more recommendations of places. The advantages of using a RS in transportation are to provide recommendations to users regardless of their location and time due to mobile devices. Mobiles have provided the possibility to access valuable information about a user's physical location. Even though mobile devices are becoming the main platform for information access, users may have difficulty finding recommendations using small-screen devices. The more users have to scroll pages, the lower the chance for an item to be found. Another challenge of applying RSs in transportation is data sparsity, as most of these systems rely on locations where a user has visited. The number of physical locations that users visit is limited, resulting in a sparse user–item matrix. Also, users may visit locations where they have never been before, making it challenging to apply collaborative filtering methods based on the user's history. Social networks are widely used in recommending locations. Reference [11] describes location-based social network RSs (LBSN) based on the relationship between user, activities, and locations. Historical information of users' location differentiates LBSN recommenders from traditional ones. Also, social networks have facilitated the use of LBSN recommenders. They are used to provide location, activity, or friend recommendations to people. Activity is considered to give a location's suggestions to satisfy a user's demand for activities such as sports, museums, restaurants, etc. For location RSs, some researchers have considered users' temporal attributes [63], and some others relied on both temporal and spatial attributes [64]. We have categorized applications of RSs in transportation into:

1. Recommending trip
2. Recommending path

3. Recommending popular activities in a location
4. Recommending popular locations (restaurants)
5. Recommending transporters (goods transporter, bus lines, drivers)

Future directions in transportation provide recommendations to a group, which is called group discovery. Users' location data can be used in clustering algorithms for this purpose.

5.3. Recommendation Systems in the e-Health Domain

E-health and medical decisions are considered for RSs' research, aiming to help medical professionals take fast and proper medical decisions. In [6], researchers proposed a RS to recommend medical advice for cardiovascular disease patients. They tackled the problems in traditional collaborative RSs such as scalability and sparsity, and they developed a new technique by applying clustering and sub-clustering methods. The authors used k-means for clustering, and they evaluated their model using precision, recall, and MAE (Mean Absolute Error). In [65] Weighted hybrid recommendation filtering approach was adopted along with an autonomic evaluation of patients' needs to propose personalized services tackling the patients' mental health. They used the self-questionnaires method to form users' profiles. In [66], an order RS for clinics was proposed. The system predicts the order of contents for each appointment and provides recommendations for providers who like to place the order. The authors used data from five outpatient clinics, and they aimed to enhance order effectiveness. The researchers also used medical heterogeneous records and data sources in [67] to develop a RS that recommends standard treatment plans for given symptoms. In [68], the authors considered the increase in personal data acquisition and mobile health systems. They developed a fuzzy optimization model to enhance a mobile wellness recommender by incorporating various imprecision levels such as fuzzy, crisp, interval-valued fuzzy parameters. They evaluated their proposed model using accuracy, specificity, and sensitivity (i.e., true negative rate). The authors in [69] developed an e-health collaborative-based RS using deep learning. They applied CNN (convolutional neural network) algorithm, and they evaluated their recommender using precision, recall, MAE, and RMSE values. The authors used sentiment analysis to obtain patients' opinions, and they kept the patients' privacy preserved. The authors in [70] developed a tailored hybrid RS incorporating demographic, utility-based, and content-based filtering techniques. They aimed to help smokers quit smoking by sending them motivational messages seeking to change their behavior. They evaluated their model using the F-measure, MAE, and the hit rate. The authors in [71] tackled the patients' dietary needs by proposing a novel method to recommend food, based on the patients' medical history. They also included other features such as age, weight, gender, calories, protein, and fat. They combined deep learning and machine learning methods—such as naïve Bayes, recurrent neural network, and Long Short Term-Memory (LSTM)—and applied them on a 30 patients' collected dataset. The emotional health-related RS is also proposed in [72]. The authors used a knowledge-based filtering recommender to suggest remedies for the patients, such as music therapy, art therapy, naturopathy, etc. that can enhance their given state of health. In [73], the authors used a hybrid filtering approach, combining context-based with collaborative filtering techniques to discover a cohort of rare diseases. They applied the model to an Alzheimer's disease dataset.

5.4. Recommendation Systems in Agriculture

In Agriculture, RSs have a significant impact on managing and using the resources efficiently, such as fertilizers, agrochemicals, irrigation. In [7], a fertilizer RS was developed to enrich the soil and increase its productivity. The authors used an ensemble classifier to suggest crops and evaluated their system using response time and accuracy measures. The issue of pests in crops was addressed and tackled in [74], where the researchers developed a RS that identifies the pests and recommends suitable treatments. In [75], the authors developed a web collaborative-based RS to answer the farmers' inquiries and update them with recent agriculture trends. They used a dataset from a call center that answers the farmers' queries by phone calls. In [76], the authors used the Apriori model and hybrid filtering model to build a web-based RS. They used Apriori in analyzing the data based on frequent items, and they recommended items to the users based on both the historical purchases and bestselling Agri-products. In [50], a collaborative based RS was proposed, where it suggests to the farmers the suitable crop according to the farmers' locations and the weather conditions. The authors used cosine similarity, and their dataset was based on four hundred farmers' information. In [77], the authors utilized an ensemble model with majority voting as they incorporated K-nearest and Naïve Bayes to develop a crop RS. The authors in [78] took the weather conditions as an input in their RS model. They proposed a hybrid filtering based RS that recommends the best crop produced for specific weather conditions. The authors applied fuzzy c means, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) for weather prediction and evaluated their model using accuracy measures. In [79], the authors aimed to improve crop productivity by proposing a RS based on ensemble technique where they incorporated different models such as (naïve Bayes, random forest, and Linear SVM). Their proposed model recommends crop types based on the input soil dataset. They evaluated their model using the overall average accuracy measure.

5.5. Recommendation Systems in Media and Beyond

The technological developments and changes in media and the increasing number of people visiting cultural places have led to an increase in various cultural items and offers. Therefore, visitors are bombarded with the information, making it difficult for them to find their interests. Thus, recommendation systems have become a vital tool to provide suggestions that ease the information overload in this area. In [80], social information as artwork attributes (e.g., type, date of creation, artist, and technical material) and user experience in a real art event is used to find the relation between users. It was shown that combining three recommender systems, including content-based, social-based, and context-based performs well in the cultural heritage domain. Museums, as one of the most important types of cultural heritages, in [81], a survey on intelligent recommender systems for museums and showed the use of users' location information and social interactions. Mobile apps have also contributed a lot in using RS systems in the cultural heritage domain. A smart search museum mobile application is used in [82] with context-aware and hybrid RSs. They have introduced a big data architecture which catches users' information—such as tastes, preferences, behaviors, needs, position, etc.—from social networks to provide recommendations.

In addition to the adoption of RSs in the cultural heritage domain, RSs are expanded to multimedia content in text, image, video, audio, etc. to help users find their favorite multimedia content. Users' multimedia data in social media are used in [83], where users' preferences, opinions, behaviors, and feedback are incorporated using metadata, textual comments, activity logs of users on a site, and ratings. They also introduced a new RS for big data application. Video RSs are widely used, especially on Netflix and YouTube. RSs rely on metadata to provide a personalized recommendation of movies; in [84] a RS based on users' preferences using machine learning is introduced. Another multimedia RS model is presented in [85], which uses social relationship mining methods and movies' metadata, users' comments, and conversation content. The recommendation results have been enhanced by applying sentiment analysis, the SVM model, and Word2Vec-based social relationships. Open social networks (OSN) play a critical role in providing personalized recommendations to users by exploiting the data and feedback retrieved from social relationships and consumer profiles on the network. In [86], the authors used the lexical analysis of Twitter's data and generated ranking scores to identify similar users. In [87], a novel music recommendation system based on the users' behaviors and their personality traits extracted from OSN is introduced. They embedded their findings with a content-based filtering approach to increase the recommender accuracy. In [88], the authors developed a diffusion interference method based on the OSN recommendation system. In [89], a collaborative and user-centered technique that exploits the users' relationships and interactions with the generated multimedia content is presented. The model consists of three stages—including data prefiltering, ranking, and similarity calculation—where they used a subset of the Yahoo Flicker 100 Million multimedia dataset. In [90], a new trust-based privacy-preserving framework for decentralized friend recommendation in OSNs (ARMOR), which utilizes OSN users' social trust relationships to generate a friend recommendation in a privacy persevering fashion, is presented. They adopted a real dataset that contains Facebook networks for 100 universities.

Table 2 shows a summary of business adoption of various recommendation systems in the four categories and their references.

Table 2. Business adoption of Recommendation Systems (RSs) in five application areas.

Area	Application	Reference
e-commerce	Items recommendations to buyers	[4,5]
	Movie or video recommendations	[43]
Transportation	Path Recommendation for transporting goodOr passengers	[8,39,49]
	Recommendations to Tourists	[50–52]
	Venue recommendation	[53–55]
e-health	Medical advice or treatment plan recommendation	[6,46,63,64]
	Recommending Personalized services to patients	[44]
	Appointments recommendation to clinicians	[45]
	Health recommendations in mobile systems	[59]
	Healthy behavioral recommendations	[61]
	Diet recommendation	[62]
Agriculture	Fertilizer recommendation to farmers	[7]
	Crops issue recommendation	[47]
	Assisting farmers inquiries	[48]
	Agricultural products recommendation	[65]
	Crop cultivation suggestion	[40,66–68]
Media	Event recommendations	[80]
	Museum recommendations	[81,82]
	Multimedia recommendations	[83–85]
	Open Social Networks recommendations	[86–90]

6. Conclusions and Future Directions

In this paper, we presented a detailed survey of RS that introduces different types of RSs as collaborative filtering, content-based, demographic-based, utility-based, knowledge-based, and hybrid-based. Different combination strategies of hybrid-based systems are also presented and categorized into weighted, mixed, switching, feature combination, feature augmentation, cascade, and meta-level. We presented four main challenges that affect the performance of a recommendation system, including cold-start, data sparsity, scalability and diversity, and metrics used to evaluate its performance. Furthermore, we show how recommendation systems have been adopted in e-commerce and various domains such as transportation, E-health, agriculture, and media. We rely on a substantial body of research to describe multiple applications in each field. We can conclude that: (1) The emergence need of more robust recommendation algorithms has led to wider applications of RSs. For instance, highly accurate deep learning methods are resulted in applying RSs in the health industry. (2) Technology advancements and smartphones have facilitated the use of RSs in daily life. For example, RSs recommend a path to drivers and passengers or the ones assisting farmers' tasks. Therefore, the effectiveness of RSs is verified in numerous areas, and they are increasingly popular. Future research includes incorporating technological opportunities such as blockchain, IoT, and RSs. With the growing number of deep learning-based RSs and the number of users and items in online platforms, new approaches that scale well with massive datasets are a future direction to add.

Author Contributions: Conceptualization, Z.F. and R.K.; methodology, Z.F., M.E., D.N., A.I., and R.K.; resources, Z.F., M.E., D.N., A.I., and R.K.; writing—original draft preparation, Z.F., M.E., D.N., A.I., and R.K.; writing—review and editing, Z.F., M.E., D.N., A.I., and R.K; visualization, Z.F., M.E., D.N., A.I., and R.K.; supervision, R.K.; project administration, R.K.; funding acquisition, R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ryerson University, Start-up Fund and the APC was funded by the same fund.

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

References

1. Beel, J.; Langer, S.; Genzmehr, M.; Gipp, B.; Breitinger, C.; Nürnberger, A. Research paper recommender system evaluation: A quantitative literature survey. In Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, Hong Kong, China, 12 October 2013; pp. 15–22.
2. Resnick, P.; Varian, H.R. Recommender systems. *Commun. ACM* **1997**, *40*, 56–59. [CrossRef]
3. Ricci, F.; Rokach, L.; Shapira, B. Introduction to recommender systems handbook. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 1–35.
4. Schafer, J.B.; Konstan, J.A.; Riedl, J. E-commerce recommendation applications. *Data Min. Knowl. Discov.* **2001**, *5*, 115–153. [CrossRef]
5. Smith, B.; Linden, G. Two decades of recommender systems at Amazon.com. *IEEE Internet Comput.* **2017**, *21*, 12–18. [CrossRef]
6. Mustaqeem, A.; Anwar, S.M.; Majid, M. A modular cluster-based collaborative recommender system for cardiac patients. *Artif. Intell. Med.* **2019**, *102*. [CrossRef]
7. Archana, K.; Saranya, K.G. Crop Yield Prediction, Forecasting and Fertilizer Recommendation using Voting Based Ensemble Classifier. *SSRG Int. J. Comput. Sci. Eng.* **2020**, *7*. [CrossRef]
8. Gheraibia, M.Y.; Gouin-Vallerand, C. Intelligent mobile-based recommender system framework for smart freight transport. In Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good, Valencia, Spain, 25–27 September 2019; pp. 219–222.
9. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–132. [CrossRef]
10. Yang, X.; Guo, Y.; Liu, Y.; Steck, H. A survey of collaborative filtering based social recommender systems. *Comput. Commun.* **2014**, *41*, 1–10. [CrossRef]
11. Ding, Z.; Li, X.; Jiang, C.; Zhou, M. Objectives and state-of-the-art of location-based social network recommender systems. *ACM Comput. Surv. CSUR* **2018**, *51*, 1–28. [CrossRef]
12. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [CrossRef]
13. Ricci, F.; Rokach, L.; Shapira, B. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015. [CrossRef]
14. Aggarwal, C.C. An Introduction to Recommender Systems. In *Recommender Systems*; Springer: Cham, Switzerland, 2016. [CrossRef]
15. Kardan, A.A.; Ebrahimi, M. A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Inf. Sci.* **2013**, *219*, 93–110. [CrossRef]
16. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. TOIS* **2004**, *22*, 5–53. [CrossRef]
17. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In *WWW '01, Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001*; Association for Computing Machinery: New York, NY, USA, 2001; Volume 1, pp. 285–295.

18. Gong, S.; Cheng, G. Mining user interest change for improving collaborative filtering. In Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application, Shanghai, China, 21–22 December 2008; Volume 3, pp. 24–27.
19. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324.
20. Vozalis, M.; Margaritis, K.G. Collaborative filtering enhanced by demographic correlation. In Proceedings of the AIAI Symposium on Professional Practice in AI, Part of the 18th World Computer Congress, Toulouse, France, 22–27 August 2004.
21. Al-Shamri, M.Y.H. User profiling approaches for demographic recommender systems. *Knowl. Based Syst.* **2016**, *100*, 175–187. [CrossRef]
22. Deng, F. Utility-based recommender systems using implicit utility and genetic algorithm. In Proceedings of the 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC-15), Shenyang, China, 1–3 April 2015; Atlantis Press: Amsterdam, The Netherlands, 2015.
23. Burke, R. Integrating knowledge-based and collaborative-filtering recommender systems. In Proceedings of the Workshop on AI and Electronic Commerce, Orlando, FL, USA, 18–22 July 1999; pp. 69–72.
24. Burke, R. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*; CRC Press: Boca Raton, FL, USA, 2000; Volume 69, (Suppl. 32), pp. 175–186.
25. Lillegraven, T.N.; Wolden, A.C. Design of a Bayesian Recommender System for Tourists Presenting a Solution to the Cold-Start User Problem. Master's Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2010.
26. Schein, A.I.; Popescul, A.; Ungar, L.H.; Pennock, D.M. Methods and metrics for cold-start recommendations. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, 11–15 August 2002; Association for Computing Machinery: New York, NY, USA, 2002.
27. Al-Bakri, N.F.; Hashim, S.H. Reducing data sparsity in recommender systems. *Al Nahrain J. Sci.* **2018**, *21*, 138–147. [CrossRef]
28. Chen, Y.; Wu, C.; Xie, M.; Guo, X. Solving the sparsity problem in recommender systems using association retrieval. *J. Comput.* **2011**, *6*, 1896–1902. [CrossRef]
29. Massa, P.; Bhattacharjee, B. Using trust in recommender systems: An experimental analysis. In Proceedings of the International Conference on Trust Management (iTrust), Oxford, UK, 29 March–1 April 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 221–235.
30. Guo, G. Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems. In Proceedings of the 7th ACM conference on Recommender systems, Hong Kong, China, 12–16 October 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 451–454.
31. O'Donovan, J.; Smyth, B. Trust in recommender systems. In Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, CA, USA, 10–13 January 2005; Association for Computing Machinery: New York, NY, USA, 2005; pp. 167–174.
32. Guo, G.; Zhang, J.; Thalmann, D. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowl. Based Syst.* **2014**, *57*, 57–68. [CrossRef]
33. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.* **2009**, *10*, 623–656.
34. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Incremental singular value decomposition algorithms for highly scalable recommender systems. In Proceedings of the Fifth International Conference on Computer and Information Science, Dhaka, Bangladesh, 27–28 December 2002; Volume 27, p. 28.
35. Sarwar, B.M.; Karypis, G.; Konstan, J.; Riedl, J. Recommender systems for large-scale ecommerce: Scalable neighborhood formation using clustering. In Proceedings of the Fifth International Conference on Computer and Information Technology, Shanghai, China, 21–23 September 2005; Volume 1, pp. 291–324.
36. Zhou, T.; Kuscsik, Z.; Liu, J.; Medo, M.; Wakeling, J.R.; Zhang, Y. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 4511–4515. [CrossRef] [PubMed]
37. Manikanta, B.K. Tackling the Problems of Diversity in Recommender Systems. Ph.D. Thesis, Kansas State University, Manhattan, KS, USA, 2010.

38. Bortko, K.; Bartków, P.; Jankowski, J.; Kuras, D.; Sulikowski, P. Multi-criteria Evaluation of Recommending Interfaces towards Habituation Reduction and Limited Negative Impact on User Experience. *Procedia Comput. Sci.* **2019**, *159*, 2240–2248. [CrossRef]
39. Hu, R.; Pu, P. Enhancing recommendation diversity with organization interfaces. In Proceedings of the 16th International Conference on Intelligent User Interfaces, Palo Alto, CA, USA, 13–16 February 2011; pp. 347–350.
40. Lee, J.; Ahn, J.H.; Park, B. The effect of repetition in internet banner ads and the moderating role of animation. *Comput. Hum. Behav.* **2015**, *46*, 202–209. [CrossRef]
41. Jankowski, J.; Hamari, J.; Watróbski, J. A gradual approach for maximising user conversion without compromising experience withhigh visual intensity website elements. *Internet Res.* **2019**, *29*, 194–217. [CrossRef]
42. Vaishnavi, S.; Jayanthi, A.; Karthik, S. Ranking technique to improve diversity in recommender systems. *Int. J. Comput. Appl.* **2013**, *68*, 2.
43. Ge, M.; Delgado-Battenfeld, C.; Jannach, D. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 257–260.
44. Gunawardana, A.; Shani, G. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.* **2009**, *10*, 2935–2962.
45. del Olmo, F.H.; Gaudioso, E. Evaluation of recommender systems: A new approach. *Expert Syst. Appl.* **2008**, *35*, 790–804. [CrossRef]
46. Cremonesi, P.; Turrin, R.; Lentini, E.; Matteucci, M. An evaluation methodology for collaborative recommender systems. In Proceedings of the 2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution, Florence, Italy, 17–19 November 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 224–231.
47. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* **1982**, *143*, 29–36. [CrossRef]
48. Chen, T.Y.; Kuo, F.-C.; Merkel, R. On the statistical properties of testing effectiveness measures. *J. Syst. Softw.* **2006**, *79*, 591–601. [CrossRef]
49. Rodriguez-Carrion, A.; Garcia-Rubio, C.; Campo, C.; Cortés-Martín, A.; Garcia-Lozano, E.; Noriega-Vivas, P. Study of LZ-based location prediction and its application to transportation recommender systems. *Sensors* **2012**, *12*, 7496–7517. [CrossRef]
50. Madhusree, K.; Rath, B.K.; Mohanty, S.N. Crop Recommender System for the Farmers using Mamdani Fuzzy Inference Model. *Int. J. Eng. Technol.* **2018**, *7*, 277–280.
51. Sezgin, E.; Sevgi, Ö. A systematic literature review on Health Recommender Systems. In Proceedings of the 2013 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–4.
52. Zhang, M.; Yi, C.; Xiaohong, Z.; Junyong, L. Study on the recommendation technology for tourism information service. In Proceedings of the 2009 Second International Symposium on Computational Intelligence and Design, Changsha, China, 12–14 December 2009; IEEE: Piscataway, NJ, USA, 2009; Volume 1, pp. 410–415.
53. Panagiotis, S.; Nanopoulos, A.; Manolopoulos, Y. MoviExplain: A recommender system with explanations. In Proceedings of the Third ACM Conference on Recommender Systems, New York, NY, USA, 22–25 October 2009; pp. 317–320.
54. Geuens, S.; Coussement, K.; De Bock, K.W. A framework for configuring collaborative filtering-based recommendations derived from purchase data. *Eur. J. Oper. Res.* **2017**, *265*, 208–218. [CrossRef]
55. Sulikowski, P.; Zdziebko, T. Deep Learning-Enhanced Framework for Performance Evaluation of a Recommending Interface with Varied Recommendation Position and Intensity Based on Eye-Tracking Equipment Data Processing. *Electronics* **2020**, *9*, 266. [CrossRef]
56. Srifi, M.; Oussous, A.; Ait Lahcen, A.; Mouline, S. Recommender Systems Based on Collaborative Filtering Using Review Texts—A Survey. *Information* **2020**, *11*, 317. [CrossRef]
57. Szabó, R.; Farkas, K.; Ispany, M.; Benczúr, A.A.; Bátfai, N.; Jeszenszky, P.; Laki, S.; Vágner, A.; Kollár, L.; Sidló, C.; et al. Framework for smart city applications based on participatory sensing. In Proceedings of the 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, Hungary, 2–5 December 2013; pp. 295–300.

58. Ricci, F. Mobile recommender systems. *Inf. Technol. Tour.* **2011**, *12*, 205–231. [CrossRef]
59. Park, M.H.; Hong, J.; Cho, S. Location-based recommendation system using bayesian user's preference model in mobile devices. In Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing, Hong Kong, China, 11–13 July 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 1130–1139.
60. Brown, B.; Chalmers, M.; Bell, M.; Hall, M.; MacColl, I.; Rudman, P. Sharing the square: Collaborative leisure in the city streets. In Proceedings of the Ninth European Conference on Computer-Supported Cooperative Work (ECSW), Paris, France, 18–22 September 2005; Springer: Dordrecht, The Netherlands, 2005; pp. 427–447.
61. Yin, H.; Sun, Y.; Cui, B.; Hu, Z.; Chen, L. LCARS: A location-content-aware recommender system. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–13 August 2013; pp. 221–229.
62. Horozov, T.; Narasimhan, N.; Vasudevan, V. Using location for personalized POI recommendations in mobile environments. In Proceedings of the International Symposium on Applications and the Internet (SAINT'06), Phoenix, AZ, USA, 23–27 January 2006; IEEE: Piscataway, NJ, USA, 2006; p. 6.
63. Gao, H.; Tang, J.; Liu, H. Exploring temporal effects for location recommendation on location based social networks. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 93–100.
64. Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; Magnenat-Thalmann, N. Time-aware point-of-interest recommendation. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 28 July–1 August 2013; pp. 363–372.
65. Hamid, K. Personalized Healthcare System Based on Ontologies. In Proceedings of the International Conference on Advanced Intelligent Systems for Sustainable Development (AI2SD '2018), Tangiers, Morocco, 12–14 July 2018; Volume 4, p. 185.
66. Sung, Y.-S.; Dravenstott, R.W.; Darer, J.D.; Devapriya, P.D.; Kumara, S. SuperOrder: Provider order recommendation system for outpatient clinics. *Health Inform. J.* **2019**, *26*, 999–1016. [CrossRef]
67. Frittelli, D.M.V. An Architecture for e-Health Recommender Systems Based on Similarity of Patients' Symptoms. In *Blockchain Technology for Smart Cities*; Springer: Singapore, 2020.
68. Mezei, J.; Nikou, S. Fuzzy optimization to improve mobile health and wellness recommendation systems. *Knowl. Based Syst.* **2018**, *142*, 108–116. [CrossRef]
69. Sahoo, A.K.; Pradhan, C.; Barik, R.K.; Dubey, H. DeepReco: Deep learning based health recommender system using collaborative filtering. *Computation* **2019**, *7*, 25. [CrossRef]
70. Hors-Fraile, S.; Núñez Benjumea, F.J.; Hernández, L.C.; Ruiz, F.O.; Fernandez-Luque, L. Design of two combined health recommender systems for tailoring messages in a smoking cessation app. *arXiv* **2016**, arXiv:1608.07192.
71. Iwendi, C.; Khan, S.; Anajemba, J.H.; Bashir, A.K.; Noor, F. Realizing an Efficient IoMT-Assisted Patient Diet Recommendation System through Machine Learning Model. *IEEE Access* **2020**, *8*, 28462–28474. [CrossRef]
72. Amelie Gyrard, A.S. IAMHAPPY: Towards an IoT knowledge-based cross-domain well-being recommendation system for everyday happiness. *Smart Health* **2020**, *15*, 100083. [CrossRef]
73. Almeida, J.R.; Monteiro, E.; Silva, L.B.; Sierra, A.P.; Oliveira, J.L. A Recommender System to Help Discovering Cohorts in Rare Diseases. In Proceedings of the 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS), Rochester, MN, USA, 28–30 July 2020.
74. Lacasta, J.; Lopez-Pellicer, F.J.; Espejo-García, B.; Nogueras-Iso, J.; Zarazaga, F.J.-S. Agricultural recommendation system for crop protection. *Comput. Electron. Agric.* **2018**, *152*, 82–89. [CrossRef]
75. Jaiswal, S.; Kharade, T.; Kotambe, N.; Shinde, S. Collaborative Recommendation System For Agriculture Sector. *ITM Web Conf.* **2020**, *32*, 03034. [CrossRef]
76. Santosh Kumar, M.B.; Balakrishnan, K. Development of a Model Recommender System for Agriculture Using Apriori Algorithm. In *Cognitive Informatics and Soft Computing*; Springer: Singapore, 2019; pp. 153–163.
77. Pudumalar, S.; Ramanujam, E.; Harine Rajashree, R.; Kavya, C.; Kiruthika, T.; Nisha, J. Crop recommendation system for precision agriculture. In Proceedings of the 2016 Eighth International Conference on Advanced Computing (ICoAC), Chennai, India, 19–21 January 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 32–36.
78. Bangaru Kamatchi, R.S. Improvement of Crop Production Using Recommender System by Weather Forecasts. *Procedia Comput. Sci.* **2019**, *165*, 724–732. [CrossRef]

79. Kulkarni, N.H.; Srinivasan, G.N.; Sagar, B.M.; Cauvery, N.K. Improving Crop Productivity through a Crop Recommendation System Using Ensembling Technique. In Proceedings of the 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 20–22 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 114–119.
80. Hong, M.; Jung, J.J.; Piccialli, F.; Chianese, A. Social recommendation service for cultural heritage. *Pers. Ubiquitous Comput.* **2017**, *21*, 191–201. [CrossRef]
81. Pavlidis, G. Recommender systems, cultural heritage applications, and the way forward. *J. Cult. Herit.* **2019**, *35*, 183–196. [CrossRef]
82. Su, X.; Sperlì, G.; Moscato, V.; Picariello, A.; Esposito, C.; Choi, C. An edge intelligence empowered recommender system enabling cultural heritage applications. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4266–4275. [CrossRef]
83. Yang, J.; Wang, H.; Lv, Z.; Wei, W.; Song, H.; Erol-Kantarci, M.; Kantarci, B.; He, S. Multimedia recommendation and transmission system based on cloud platform. *Future Gener. Comput. Syst.* **2017**, *70*, 94–103. [CrossRef]
84. Mwinyi, I.H.; Narman, H.S.; Fang, K.; Yoo, W. Predictive self-learning content recommendation system for multimedia contents. In Proceedings of the 2018 Wireless Telecommunications Symposium (WTS), Phoenix, AZ, USA, 18–20 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
85. Baek, J.-W.; Chung, K.-Y. Multimedia recommendation using Word2Vec-based social relationship mining. *Multimed. Tools Appl.* **2020**, 1–17. [CrossRef]
86. Aysha, K.; Ali, R. Recommender System Based on OSN Data Analytics. In *Information and Communication Technology for Intelligent Systems, Proceedings of ICTIS 2018, Padang, Indonesia, 25–26 July 2018*; Springer: Singapore, 2018; Volume 2, p. 2.
87. Moscato, V.; Picariello, A.; Sperli, G. An Emotional Recommender System for music. *IEEE Intell. Syst.* **2020**, 1. [CrossRef]
88. Yang, X.; Dong, M.; Chen, X.; Ota, K. Recommender System-Based Diffusion Inferring for Open Social Networks. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 24–34. [CrossRef]
89. Sperlì, G.; Amato, F.; Mercorio, F.; Mezzanzanica, M. A Social Media Recommender System. *Int. J. Multimed. Data Eng. Manag.* **2018**, *9*, 36–50. [CrossRef]
90. Ma, X.; Jianfeng, M.; Li, H.; Jiang, Q.; Gao, S. ARMOR: A trust-based privacy-preserving framework for decentralized friend recommendation in online social networks. *Future Gener. Comput. Syst.* **2017**, *79*, 82–94. [CrossRef]

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