An Organized Analysis on Effective Utilization of Machine Learning Algorithms in Recommender Systems

Mr. Kommaddi Tharun[1]
[1] Under Graduate, Dept of Computer Science and Engineering, JNTUACEP, Pulivendula.

Abstract — *Recommender systems use algorithms to give clients item or administration recommendations. As of late, these systems have been utilizing machine learning algorithms from the field of artificial knowledge. Be that as it may, picking a suitable machine learning algorithm for a recommender framework is troublesome because of the quantity of algorithms portrayed in the literature. Researchers and practitioners creating recommender systems are left with little information about the momentum approaches in algorithm usage. Additionally, the improvement of recommender systems utilizing machine learning algorithms often faces issues and raises addresses that must be settled. This paper exhibits a systematic review of the literature that analyzes the utilization of machine learning algorithms in recommender systems and recognizes new research openings. The goals of this examination are to (I) distinguish slants in the utilization or research of machine learning algorithms in recommender systems; (ii) recognize open inquiries in the utilization or research of machine learning algorithms, and (iii) assist new researchers to position new research activity in this domain appropriately. The consequences of this examination distinguish existing classes of recommender systems, characterize adopted machine learning approaches, talk about the utilization of enormous data advances, recognize sorts of machine learning algorithms and their application domains, and analyzes both main and alternative performance metrics*.

Keywords—*Systematic review, recommender systems, machine learning, machine learning algorithms, application domains, performance metrics.*

I. INTRODUCTION

Recommender systems (RSs) are utilized to enable clients to discover new things or administrations, for example, books, music, transportation or even individuals, based on information about the client, or the suggested thing (Adomavicius and Tuzhilin, 2005). These systems also play an important job in basic leadership, helping clients to maximize profits (L.- S. Chen, Hsu, Chen, and Hsu, 2008) or limit dangers (Bouneffouf, Bouzeghoub, and Gancarski, 2013). Today, RSs are utilized in many information-based companies, for example, Google (J. Liu, Dolan, and Pedersen, 2010), Twitter (Ahmed et al., 2013), LinkedIn (Rodriguez, Posse, and Zhang, 2012), and Netflix (Steck, 2013). The field of RS has its beginnings in the mid-1990s with the presentation of Tapestry (Goldberg, Nichols, Oki, and Terry, 1992), the main RS.

As the RS field advanced, researchers considered the utilization of algorithms from machine learning (ML), an area of artificial insight (AI). ML has been concentrated since the late 1950s (Martens, 1959), with the rise of the field of AI. Today, there is a plethora of ML algorithms (k-nearest neighbor (Patrick and Fischer III, 1970), grouping (Jain, Murty, and Flynn, 1999), Bayes arrange (Friedman, Geiger, and Goldszmidt,1997), to name a couple of sorts), which are utilized in applications that range from vacuum cleaner robots (Burhans and Kandefer, 2004) and assistance for disabled individuals (Karimanzira, Otto,& Wernstedt, 2006) to pattern acknowledgment in images (Torralba, Fergus, and Weiss, 2008), or self-driving vehicles (Thrun, 2007). The potential application of ML algorithms is vast and the field looks extremely encouraging.

ML algorithms are being utilized in RSs to furnish clients with better recommendations. Be that as it may, the ML field does not have a clear classification plot for its algorithms, mainly because of the quantity of approaches and the variations proposed in the literature (Lv and Tang, 2011). As an outcome, it ends up troublesome and befuddling to pick a ML algorithm that accommodates one's need when building up a RS. In addition, researchers may think that its challenging to track the utilization and the patterns of ML algorithms in RSs.

One way to assist researchers and practitioners (for example soft-ware specialists and designers (Pressman, 2015; Isazadeh, 2004)) in picking which ML algorithm to use in a RS is the investigation of the RS and ML fields. Research about RSs containing ML algorithms executed in the literature can help show drifts and give a course to future examinations.

This paper gives a systematic review to investigate how ML algorithms utilized in RSs are contemplated and utilized; and what are the patterns in ML algorithm research and improvement. It is normal that, with this systematic review, researchers and practitioners can obtain more information about the RS field, and make better implementation or research choices. The goals of this examination are to (I) distinguish slants in the utilization or research of machine learning algorithms in recommender systems; (ii) recognize open inquiries in the utilization or research of machine learning algorithms; and (iii) assist new researchers to position new research activity in this domain appropri-ately. The aftereffects of this investigation recognize existing classes of recommender systems, characterize adopted machine learning approaches, talk about the utilization of huge data advancements, distinguish kinds of machine learning algorithms and their application do-mains, and analyze main and alternative performance metrics

II. RELATED WORK

This section gives an overview of the two main research fields related to this article, namely recommender systems and machine learning.

2.1 Recommender Systems

Recommender systems (RSs) utilize artificial knowledge (AI) techniques to give clients thing recommendations. For example, an online bookshop may utilize a machine learning (ML) algorithm to classify books by sort and then prescribe other books to a client purchasing a particular book. RSs were presented in 1992 when Tapestry, the principal RS, appeared. Its authors utilized the term collaborative separating to allude to the recommendation activity. This term is as yet used to classify RSs. RSs are separated into three main categories to drive the recommendations: collaborative, content-based, and mixture sifting (Adomavicius and Tuzhilin, 2005).

To start with, RSs utilizing a collaborative approach consider the client data when handling information for the recommendation. For instance, by accessing client profiles in an online music store, the RS has access to all the client data, for example, the age, nation, city, and tunes purchased. With this information, the framework can distinguish clients that share the same music inclination and then propose melodies purchased by similar clients.

Second, RSs with a substance based separating approach base their recommendations on the thing data they can access. As an example, consider a client who is searching for another PC utilizing an online store. At the point when the client peruses a particular PC (thing), the RS gathers information about that PC and searchers in a database for PCs that have similar attributes, for example, value, CPU speed, and memory capacity. The aftereffect of this search is then come back to the client as recommendations.

The third category portrays RSs that join the two past categories into a half and half separating approach, prescribing things based on the client and the thing data. For example, on a social system, a RS may prescribe profiles that are similar to the client (collaborative sifting), by comparing their interests. In a second step, the framework may consider the suggested profiles as things and therefore access their data to search for new similar profiles (content-based sifting). At last, the two arrangements of profiles are returned as recommendations.

When utilizing a collaborative or a half breed separating approach, RSs must gather information about the client so as to create recommendations. This activity can be done expressly or certainly. Unequivocal client data gathering (Sutton and Barto, 1998) happens when clients are aware they are giving their information. For instance, while enrolling for another online administration, clients usually fill in a frame that asks their name, age, and email. Other types of unequivocal client data gathering (Gemmis et al., 2011; Longo, Barrett, and Dondio, 2009) are when clients express their inclinations by rating things utilizing a numerical value or an inclination, for example, a Facebook "like." Implicit client data gathering accesses information about the client by implication. For example, when visiting an online store, the server at the online store exchanges messages with the client's PC, and based on that, the store's RS may know the program the client is utilizing, as well as the client's nation. Further developed applications screen client snaps and keystroke logs.

Other than the basic recommendation process, in which clients are given things that may be of intrigue, recommendations can be given in other ways. Trust-based recommendations (O'Donovan and Smyth, 2005) take into consideration the trust relationship that clients have between them. A trust relationship is a connection in a social system to a companion or a related association. Recommendations based on trust are worth more than those that don't have trust joins. Setting aware recommendations (Adomavicius, Mobasher, Ricci, and Tuzhilin, 2011) are based on the setting of the client. A setting is a lot of information about the present state of the client, for example, the time at the client location (morning, afternoon, evening), or their activity (inactive, running, resting). The amount of setting information to be prepared is high, making setting aware recommendations a challenging research field. Hazard aware recommendations (Bouneffouf et al., 2013) are a subset of setting aware recommendations and take into consideration a setting in which critical information is available, for example, client vital signs. It is hazard aware because a wrong choice may threaten a client's life or cause damage. A few examples are prescribing pills to be taken or stocks the client should purchase or, move.

2.2 Machine Learning

Machine Learning (ML) utilizes PCs to simulate hu-man learning and allows PCs to recognize and acquire information from the real world, and enhance performance of a few tasks based on this new information. All the more formally, ML is characterized as pursues: "A PC program is said to learn for a fact E as for some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, enhances with experience E" (R. Michalski, Carbonell, and Mitchell, 1985). Although the principal ideas of ML originated during the 1950s, ML was concentrated as a separate field during the 1990s (R. S. Michalski, Carbonell, and Mitchell, 2013). Today, ML algorithms are utilized in several areas other than software engineering, including business (Apte, 2010), advertising (Cui, Bai, Gao, and Liu, 2015) and medication (Kononenko, 2001).

Learning is the procedure of information acquisition. Humans naturally learn as a matter of fact because of their ability to reason. In contrast, PCs don't learn by reasoning, however learn with algorithms. Today, there are countless algorithms proposed in the literature. They can be classified based on the approach utilized for the learning procedure. There are four main classifications: regulated, unsupervised, semi-administered, and fortification learning.

Regulated learning (Kotsiantis, 2007; D. Zhang and Tsai, 2006) happens when algorithms are furnished with training data and right answers. The task of the ML algorithm is to learn based on the training data, and to apply the information that was gained utilizing real data. As an example consider a ML learning algorithm being utilized for book classification in a book shop. A training set (training data + answers) can be a table relating information about each book to a right classification. Here, information about each book may be title, author, or even every word a book contains. The ML algorithm learns with the training set. At the point when another book arrives at the book shop, the algorithm can classify it based on the information about book classification it has acquired.

In unsupervised learning (Celebi and Aydin, 2016), ML algorithms don't have a training set. They are given a few data about the real world and have to learn from that data all alone. Unsupervised learning algorithms are for the most part fo-cused on finding shrouded patterns in data. For example, assume that a ML algorithm has access to client profile information in a social system. By utilizing an unsupervised learning approach, the algorithm can separate clients into personality categories, for example, cordial and saved, allowing the social system company to target advertising all the more specifically at explicit gatherings of clients.

ML algorithms can also be classified as semi-regulated. Semi-regulated learning (Chapelle, Scholkopf,¨ and Zien, 2006; Xu, Mo, and King, 2012) happens when algorithms work with a training set with missing information, and still need to learn from it. An example is the point at which a ML algorithm is given motion picture ratings. Few out of every odd client rated each film and along these lines, there is some missing information. Semi-directed learning algorithms are able to learn and draw ends even with inadequate data.

Lastly, ML algorithms may have a fortification learning approach. Fortification learning (Sutton and Barto, 1998) happens when algorithms learn based on external feedback given either by a reasoning element, or the earth. This approach which they are structured. A few examples of classification can be found in (Shalev-Shwartz and Ben-David, 2013) and (Kulkarni, 2012), although the field still does not have any standard classification.

When creating RSs, software engineers must settle on the particular recommender algorithm of all those available. This decision has significant impact on the rationale of the RS, on the data that will be required from clients and recommendation things, and on performance issues. The quantity of algorithm variations and combinations in the literature makes this decision a challenging task.

This large number of recommender algorithms, which ap-pears to be constantly developing and changing, makes software designing for RSs a proceeding with challenge. Attempting to create devices to make RS advancement easier is a moving target, as new examinations must be done to watch new open issues and drifts, and further enhance the learning base.

Consequently the authors directed a systematic review to analyze the improvement of RSs containing ML algorithms. This systematic review pursues the strategies of (Kitchenham, 2004) and has, as goals, to:

1) Identify patterns in the utilization or research of ML algorithms in RSs,

2) Identify open inquiries in the utilization or research of ML algorithms,

3) Assist new researchers to position new research activity in this domain appropriately.

This systematic review has one confinement. The authors de-cided to restrict the arrangement of studies investigated to those depicting an examination or a validation consider. The main reason for this confinement is that several publications in the literature propose new algorithms that are never tried or validated. Along these lines, by including this limitation, this systematic review can analyze the performance metrics of the ML algorithms, for example, accuracy, recall, and f-measure.

To answer the principal research question, the authors investi-gated the sort of separating strategy utilized in the recommender framework being depicted in an examination. The approach to answering the second inquiry included more data. The publication proposed in the publications had its classification (directed, unsupervised, and so on), their sort (grouping, choice tree, and so on) investigated, as well as its help for circulated advances (Hadoop, MapReduce). The performance metrics that portray each ML algorithm investigated in this systematic review were analyzed. The third inquiry is answered by examining the gatherings and journals in which the investigations were distributed, and the reviews that were returned by the search question. watchwords, and attempts to discover terms that relate to the field of RS, ML, and give some indication that the proposed approach was validated. Concentrates should also contain the expression "machine learning" in the title, abstract, or watchwords. To recover thinks about that were assessed, the search question also searches for the expressions "analysis" or its equivalent words.

The quantity of concentrates to be read in the systematic review decreased from 215 to 121 when sifted by the rejection crite-ria. Fifteen of the examination sections were gathering or continuing depictions and are prohibited because they are not composed logical work. After reading the abstract of the investigations, the authors were certain that 17 thinks about were not related to the goal of this systematic review and chose to avoid them. The authors did not have access to five investigations, even after asking assistance from colleagues and visiting libraries. These investigations were then not examined in this systematic review. Two investigations were in Chinese and another one was in Japanese. Four examinations had a duplicate returned by the search string. These examinations present the same outcomes and were not checked twice. Just the original investigation was considered in this systematic review. After reading the investigations, the individuals who did not concentrate their proposal on the key research fields of this review were rejected. Besides, considers that did not explain the ML algorithm being utilized, or did not portray a validation study, or its outcomes were also rejected from this systematic review. At last, 121 primary examinations were retained and analyzed.

One last important point to make reference to is that the investigations reviewed may propose more than one ML algorithm. As an outcome, a portion of the outcomes introduced on the following chapter are centered around the quantity of studies, while others are centered around the quantity of algorithms. The 121 investigations portrayed a total of 205 ML algorithms that are either totally new, or modifications' or optimization of existing ones. Finally, algorithms can be validated in at least one application domains. This also impacts a few outcomes appeared in the following area.

III. PROPOSAL WORK

The reading process focused on finding three types of information: one that relates to the RS being described (its classification), another that relates to the ML algorithm (its type, application domain, and performance metrics), and finally information about the source of the study. Exhibiting a systematic review of the literature that analyzes the utilization of machine learning algorithms in recommender systems and recognizes new research openings

3.1 Big Data Technologies

ML algorithms, by definition, improve their performance with access to more data. Similarly, the more data that is provided to an RS, the better should be its recommendations. The evolution of technology has spawned research into new ways of handling data. One such phenomenon is called Big Data (M. Chen, Mao, & Liu, 2014), which has produced the Hadoop distributed infrastructure (Shvachko, Kuang, Radia,&Chansler, 2010) and the MapReduce programming model (Dean & Ghemawat, 2008). Because Big Data has a direct impact in RS development and ML algorithms (Leskovec, Rajaraman, & Ullman, 2014), the authors decided to look for studies that have a discussion of Big Data in the description of their proposed algorithms. Table 4.5 shows the number of studies that included Big Data in their discussion or proposals.

Among the studies that described some Big Data adaptations, Baldominos et al. (2015) used Big Data for storage. The proposed architecture that provides on-demand tools for analysis uses the storage technologies HDFS (Hadoop Distributed File System) (Shvachko et al., 2010) and HBase6 for persistence logs and structured information about the execution and predictions. Another study (Dinuzzo et al., 2011), in the health domain, uses data from distributed datasets to make predictions. The description of the Big Data technologies used in the prediction process was not the focus of the study. Lastly, Geng et al. (2016) propose a neural network-based algorithm that is applied to the image domain and, according to the authors, easily scales to large networks and for recommender systems.

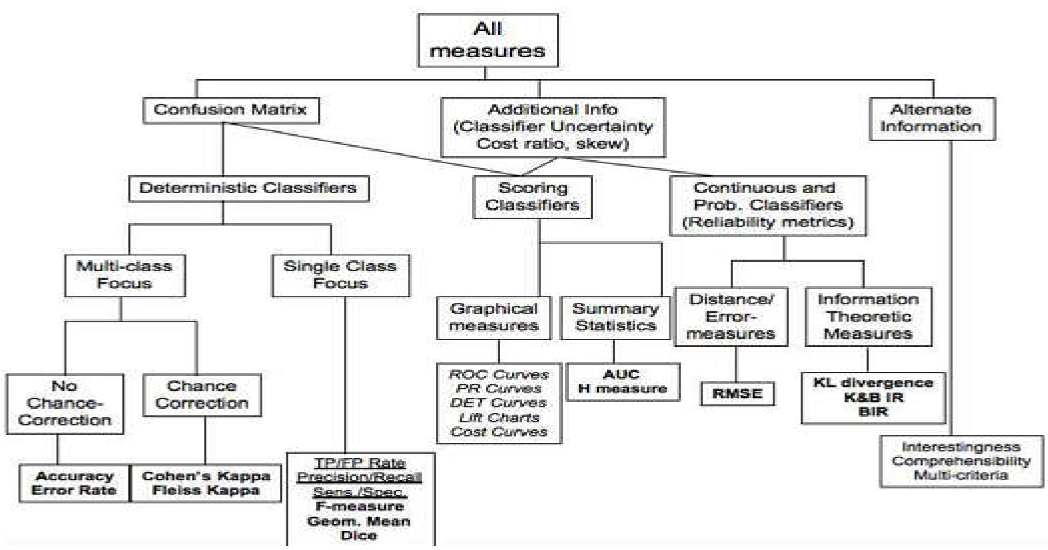
Although as mentioned earlier, it is clear that few studies had their proposals adapted for a Big Data reality, with distributed technologies or performance-optimized programming paradigms. This Big Data approach appears to represent a large research opportunity for RS development.

3.2 Application Domains

This systematic review investigates the application domains used in the studies analyzed. A primary study may propose multiple algorithms, which may be validated in many different application domains. This means that the authors may investigate the application domains on a per algorithm or a per study basis. The authors opted for the latter approach so that the number of algorithms proposed in a single study does not affect the final result of the application domain analysis.
The application domain of Movies is the one mostly used with 31 occurrences among the 121 studies. One reason for this result is the ease of access to data in the movie domain. The University of Minnesota maintains a dataset with several movie ratings, named MovieLens7, which is widely used. Another source of user ratings is the Internet Movie Database (IMDb)8, which contains millions of titles and ratings that can be used to build a testing dataset.

The social domain ranks in the second place. This domain accounts for algorithms aimed to work on social networks or applications that connect different users. This use confirms the trend of collaborative approaches in RS development with ML algorithms. The tourism and the coding domains ranked low, revealing opportunities for research since data in these domains are rich and easily accessible.

Classification of Performance Measures for ML algorithms in Recommender System development.



An overview of performance metrics (from (Japkowicz & Shah, 2011))

3.3 Alternative Performance Metrics

This section presents alternative metrics that can also be used to describe the performance of ML algorithms in RS development. These metrics are well described in another study (Gunawardana & Shani, 2015) with examples and suggested ways of capturing data and calculating results. The eight metrics are a user preference, coverage, confidence, trust, novelty, serendipity, diversity, utility, risk, robustness, privacy, and scalability. Some of them are discussed in the next few paragraphs.

User preference, as its name suggests, relates to the opinion of the user about the **recommendations** made by the RS. Users are more likely to choose approaches that predict items that match their preferences. Although the description is easy, gathering user data to achieve high user preference is not. The main method to obtain data about user preference is the use of questionnaires. The coverage metric relates to the items that can be recommended to the users that can receive recommendations. There are specific ways to calculate the coverage and one should refer to (Gunawardana & Shani, 2015) for more details.

Two additional alternative metrics are diversity and scalability. To discuss diversity, one must understand similarity, since these two concepts are antagonistic. If the results are not similar, then that means they are diverse. Lastly, scalability does not mean much to the user, but important both to research and performance. Scalability relates to how well-prepared the algorithms is to handle growth in the amount of data. Most of the time, algorithms need more memory or computation power to manipulate large amounts of data.

The difference between the “Textual” and “Numeric” entries in the table is because that discussion can be in the written form, with considerations or suppositions, or it can be based on a formula. The last column shows the studies that discussed the algorithms. The difference between the number of algorithms presented in the third column and the number of studies of the fourth column exists because a study may propose more than one algorithm.
Currently, recommender systems (RS) are widely used in e-commerce, social networks, and several other domains. Since the introduction of RSs in mid-1990s, research in RSs has been evolving. One progressive step in RS history is the adoption of machine learning (ML) algorithms, which allow computers to learn based on user information and to personalize recommendations further. Machine learning is an Artificial Intelligence (AI) research field that encompasses algorithms whose goal is to predict the outcome of data processing. ML has made major breakthroughs in the fields of image recognition, search engines, and security. However, the ML field has several algorithms described in the literature, with varied characteristics. The literature lacks a classification

IV. CONCLUSION

Right now, recommender systems (RS) are generally utilized in online business, social systems, and several other domains. Since the presentation of RSs in mid 1990s, research in RSs has been developing. One dynamic advance in RS history is the adoption of machine learning (ML) algorithms, which allow PCs to learn based on client information and to personalize recommendations further. Machine learning is an Artificial Intelligence (AI) research field that encompasses algorithms whose goal is to anticipate the result of data handling. ML has made major breakthroughs in the fields of image acknowledgment, search motors, and security. Be that as it may, the ML field has several algorithms portrayed in the literature, with varied characteristics.

V. REFERENCES

1. Adomavicius, G., Mobasher, B., Ricci, F., & Tuzhilin, A. (2011). Context-aware recommender systems. AI Mag-azine, 32(3), 67–80.

2. Adomavicius, G. & Tuzhilin, A. (2005). Toward the next gen-eration of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734–749. doi:10.1109/TKDE.2005.99

3. Bagchi, S. (2015). Performance and quality assessment of sim-ilarity measures in collaborative filtering using mahout. (Vol. 50, pp. 229–234). doi:10.1016/j.procs.2015.04.055

4. Baldominos, A., Albacete, E., Saez, Y., & Isasi, P. (2015). A scalable machine learning online service for big data real-time analysis. doi:10.1109/CIBD.2014.7011537

5. Banerjee, S., Bhowmick, P., Mukherjee, D., & Misra, P. (2012). Experiments on synopsis-based tv program rec-ommendation. (pp. 258–263).

6. Bar, A., Rokach, L., Shani, G., Shapira, B., & Schclar, A. (2013). Improving simple collaborative filtering models using ensemble methods. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial

7. Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filter-ing. In Proceedings of the fourteenth conference on un-certainty in artificial intelligence (pp. 43–52). UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc.

8. Caraballo, A., Arruda, J., N.M., Nunes, B., Lopes, G., & Casanova, M. (2014). Trtml - a tripleset recommen-dation tool based on supervised learning algorithms. Lecture

9. Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8798, 413–417. doi:10.1007/ 978-3-319-11955-7 58

10. Das, S., Moore, T., Wong, W.-K., Stumpf, S., Oberst, I., McIntosh, K., & Burnett, M. (2013). End-user feature labeling: supervised and semi-supervised approaches based on locally-weighted logistic regression. Artificial Intelligence, 204, 56–74. doi:10.1016/j.artint.2013.08. 003

11. Elmongui, H., Mansour, R., Morsy, H., Khater, S., El-Sharkasy, A., & Ibrahim, R. (2015). Trupi: twitter rec-ommendation based on users' personal interests. Lecture

12. Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9042, 272–284. doi:10 . 1007 / 978 - 3 - 319-18117-2 20

13. Fan, T.-K. & Chang, C.-H. (2010). Learning to predict ad clicks based on boosted collaborative filtering. (pp. 209– 216). doi:10.1109/SocialCom.2010.37

14. Gemmis, M., Iaquinta, L., Lops, P., Musto, C., Narducci, F.,&Semeraro, G. (2011). Learning preference models in recommender systems. doi:10.1007/978- 3- 642- 14125-6

15. Geng, X., Zhang, H., Bian, J., & Chua, T.-S. (2016). Learning image and user features for recommendation in social networks. (Vol. 11-18-December-2015, pp. 4274–4282). doi:10.1109/ICCV.2015.486

16. Hernandez´ del Olmo, F., Gaudioso, E., & Martin, E. (2009). The task of guiding in adaptive recommender systems. Expert Systems with Applications, 36(2 PART 1), 1972– 1977. doi:10.1016/j.eswa.2007.12.070

17. Islam, M., Ding, C., & Chi, C.-H. (2015). Personalized recommender system on whom to follow in twitter. (pp. 326–333). doi:10.1109/BDCloud.2014.84

18. Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: a review. ACM Computing Surveys, 31(3), 264–323. doi:10.1145/331499.331504

19. Kao, C.-Y. & Fahn, C.-S. (2013). A multi-stage learning framework for intelligent system. Expert Systems with Applications, 40(9), 3378–3388. doi:10 . 1016 / j . eswa . 2012.12.046

20. Kulkarni, S. (2012). Machine learning algorithms for problem solving in computational applications: intelligent tech-niques. IGI Global. doi:10.4018/978-1-4666-1833-6

21. Lahlou, F., Benbrahimand, H., Mountassir, A., & Kassou, I. (2013). Context extraction from reviews for context aware recommendation using text classification tech-niques. doi:10.1109/AICCSA.2013.6616512

22. Murfi, H. & Obermayer, K. (2009). A two-level learning hierarchy of concept based keyword extraction for tag recommendations. (Vol. 497, pp. 201–214).

23. Pressman, R. S. (2015). Software engineering: a practitioner's approach (8th). McGrawHill Higher Education.

24. Pronoza, E., Yagunova, E., & Volskaya, S. (2016). Aspect-based restaurant information extraction for the recom-mendation system. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intel-ligence and Lecture Notes in Bioinformatics), 9561, 371–385. doi:10.1007/978-3-319-43808-5 28

25. Rubens, N., Elahi, M., Sugiyama, M., & Kaplan, D. (2015). Active learning in recommender systems. Springer US. doi:10.1007/978-1-4899-7637-6 24

26. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Anal-ysis of recommendation algorithms for e-commerce. In Proceedings of the 2nd acm conference on electronic commerce (pp. 158–167). EC '00. Minneapolis, Min-nesota, USA: ACM. doi:10.1145/352871.352887

27. Szymanski,´ J. & Rzeniewicz, J. (2016). Identification of cat-egory associations using a multilabel classifier. Expert Systems with Applications, 61, 327–342. doi:10.1016/j. eswa.2016.05.039

28. Taghipour, N., Kardan, A., & Ghidary, S. (2007). Usage-based web recommendations: a reinforcement learning approach. (pp. 113–120). doi:10.1145/1297231.1297250

29. Takacs,´ G., Pilaszy,´ I., Nemeth,´ B., & Tikk, D. (2008). Matrix factorization and neighbor based algorithms for the netflix prize problem. (pp. 267–274). doi:10.1145/1454008.1454049

30. Verma, C., Hart, M., Bhatkar, S., Parker-Wood, A., & Dey, S. (2016). Improving scalability of personalized recom-mendation systems for enterprise knowledge workers. IEEE Access, 4, 204–215. doi:10.1109/ACCESS.2015. 2513000