{tag} {/tag}International Journal of Computer Applications
Foundation of Computer Science (FCS), NY, USA

Volume 179
-
Number 46

Year of Publication: 2018

Authors:

E. Uko Okon, B. O. Eke, P. O. Asagba

10.5120/ijca2018917193

{bibtex}2018917193.bib{/bibtex}

Abstract

In e-commerce today, contents available for users to explore are overwhelming because an average ecommerce website is about seventy per cent (70%) more than a physical store in total number of users and items. Hence, the need to filter, prioritize and efficiently deliver relevant information using recommender systems. We will design and develop a recommendation model that uses object-oriented analysis and design methodology (OOADM), improved collaborative filtering algorithm and an efficient quick sort algorithm to solve these problems. This will be achieved by implementing the stated model with python model-view-controller (MVC) framework known as Django Framework. This improved system is implemented using a real-time, cloud-hosted NOSQL database called FireBase which guarantees scalability. From the results, the speed and scalability of book recommendation was improved with a performance record obtained within the range of ninety (90) to ninety-five (95) per cent using the root mean square error (RMSE) of several recommendations obtained from the system.

References

1. Akshita, J., and Smita, A. (2013). Recommender system: review. International Journal of computer application, 71(24), 38-42.
2. Burke, R. (2007). Hybrid web recommender systems. The Adaptive Web, Springer Berlin, Heidelberg, 377–408.
3. Cheng, Z., and Hurley, N. (2010). Effective diverse and obfuscated attacks on model-based recommender systems. RecSys ’09: Proceedings of the Third ACM Conference on Recommender Systems, ACM, New York, NY, USA, 141–148.
4. Desrosier, C., and Karypis, G. (2012). A comprehensive survey of neighbourhood-based recommendation methods. Department of Computer Science and Engineering. University of Minnesota, Mineapolis, USA. 5-33.
5. Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80.
6. Mahmood, T., and Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. Hypertext ACM publications, 73–82.
7. Marlin, B. (2004). Collaborative filtering: a machine learning perspective. Thes is research. Department of computer science, University of Toronto. 1-118.
8. Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In Proceedings of the eighteenth national conference on artificial intelligence (AAAI-02), Edmonton, Alberta, 187-192.
9. Onah, D. and Sinclair, J. (2015). Collaborative filtering recommendation system: a framework in massive online courses. Proceedings of the 9th International Technology, Education and Development Conference, Madrid, Spain. 1249-1257. isbn:9788460657637.
10. Resnick, P., and Sami, R. (2007). The influence limiter: provably manipulation-resistant recommender systems. RecSys ’07: Proceedings of the 2007 ACM conference on Recommender systems, ACM, New York, USA. 25–32.
11. Sarwar, B., Karypis, G., Konstan, J., and Reidi, J. (2001). Item-based collaborative filtering algorithms. GroupLens research group, Army HPC research center, University of Minnesota, Minneapolis, 1-11.
12. Simpson, J., & Weiner, E. (2016). Recommender. Oxford English Dictionary. Oxford: Oxford University Press.
13. Su, X., and Khoshgoftaar T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence. Hindawi publishing corporation. 1-19.
14. Zhao, Z. and Shang, M. (2010). User-based Collaborative Filtering Recommendation Algorithms on Hadoop. Third International Conference on Knowledge Discovery and Data Mining. IEEE journal. 478-481. doi: 10.1109/WKDD.2010.54.

Index Terms

Computer Science Algorithms

Keywords

Recommender system, collaborative filtering, recommender algorithms, collaborative filtering algorithm, machine learning algorithm, NoSQL, Firebase, scalability.