**COMP 5327**

|  |  |
| --- | --- |
| ***Library Research*** | ***Student Name : Meerim Samakova*** |
| ***Assignment Date: 11/06/2023*** | ***Student id : 10003737*** |

**The Problem:**

We run a large e-commerce website, and want to recommend products to our users. The better your recommendations, the more likely users are to buy something, making your business more successful. But how do we know what to recommend to each user?

**Original Algorithm Analysis:**

Right now, we are using a method called "Collaborative Filtering." It works like this: when a user looks at a product or buys something, the system remembers that. Then, when another user with similar tastes comes along, the system recommends the product to them. This method is quite effective for many scenarios. However, it has some limitations:

* **New Users:** For people who just signed up and haven't done much on the website yet, it's tricky to suggest things because there's not much data about their preferences.
* **Not-So-Popular Items:** Collaborative Filtering is better at suggesting popular products because they have more data associated with them. But for less popular products, it's less effective.

**Proposed Improvement:**

I propose an advanced approach that combines Collaborative Filtering with "Content-Based Filtering." This hybrid approach takes the strengths of both methods to create more accurate and versatile recommendations.

**Algorithm Implementation:**

* **Collaborative Filtering (CF):** We keep using this method because it's good at finding products that people with similar tastes like. However, we adjust it a bit.
* **Content-Based Filtering (CBF):** We introduce this new method. It involves analyzing the characteristics of products (content) and users' preferences for these characteristics. For example, if you're recommending movies, you look at the genre, actors, directors, and user preferences for these attributes.
* **Hybrid Approach:** We don't just rely on one method; we use both. This way, we can suggest products more effectively. If someone is new, we lean more on the content-based method, considering the product details like genre. But if someone has been using the platform for a while, we start using collaborative filtering more.
* **Machine Learning:** To make our recommendations even better, we can use machine learning techniques to predict what a user might like based on their previous behavior and product attributes.

**Performance Analysis:**

To see if our new system works well, we use mathematical metrics and data to check how good our recommendations are. We compare the new approach with the old one to ensure it's an improvement.

**Discussion:**

The improved recommendation system makes your platform better for users. You get more accurate recommendations, even if you're new or if you're interested in less popular products. It's like having a personal shopping assistant who really understands your tastes.

**Conclusion:**

In conclusion, this advanced recommendation system is like having a super-smart shopping assistant. It combines two powerful methods to provide you with the best suggestions. It's not just for shopping; it can work in various areas, such as suggesting movies, music, news, and more. In the future, we can make it even smarter by exploring more advanced techniques and fine-tuning the hybrid approach for the best recommendations possible.

**References:**

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009, 4.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The Adaptive Web (pp. 325-341). Springer.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (pp. 437-438).

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370.