Athanasios Farmakis dai17102

Advantages of content-based filtering

User independence: The content-based method should only analyze a user's data and profile for the recommendation, which makes the process less cumbersome. Therefore, content filtering will produce more reliable results with fewer users on the system.

Transparency: Collaborative filtering gives recommendations based on other unknown users who have the same taste as a particular user, but with content-based filtering, feature-level elements are recommended.

There is no cold start: Unlike collaborative filtering, new items can be suggested before being evaluated by a significant number of users.

Disadvantages of content-based filtering

Limited content analysis: If the content does not contain enough information to accurately distinguish objects, the recommendation itself risks being inaccurate.

Personalization: Content filtering provides a limited degree of innovation, as it must match the capabilities of a user profile with available items. In the case of object filtering, only object profiles are created and users suggest objects similar to those they rate or search for, instead of their history. A perfect content-based filtering system can suggest nothing unexpected or surprising.

Conclusions

We have learned to create a fully functional Python recommendation system with content filtering. But as we saw above, content filtering is impractical, or rather unreliable when the number of items increases along with the need for clear and differentiated descriptions.

To overcome all the issues we have previously discussed, we can apply collaborative filtering techniques, which have proven to be better and more scalable. We will work on their implementations in the upcoming parts of the series.