awsmarketplace

Advanced Analytics: **Real-time Product Recommendations**

Challenge

Customers visit your e-commerce website with an intent to purchase a specific product. They purchase that product, and then they leave your website, not knowing that there are other products that your company may offer that would also suit their needs.

Solution

Implement a recommendation engine on your website. That way, when customers visit your e-commerce website, they can view and buy products that are similar to ones that they have already purchased from your website.

Build and Run a Recommendation Engine at Any Scale

A recommendation engine is a tool that give businesses the ability to predict what customers may or may not like about their products on their web site or mobile device. Recommendation engines also help customers discover products or content that they may not otherwise consider.

Recommendation engines work in a couple of ways. One way is that it relies on the properties of the items that a user likes, which are analyzed to determine what else the user may like; or, it can rely on the likes and dislikes of other users, which the recommendation engine then uses to calculate a similarity index between users and then recommends items to them. Both of these methods can also be combined to build a more robust recommendation engine.

One of the purposes of a recommendation engine is to make interpretations on existing data to show relationships between objects. Objects can be many things like, users, items, and products. Relationships provide a degree of likeness or belonging between objects. For example, relationships can represent ratings of how much a user likes an item, or indicate if a user bookmarked a page. To make a recommendation, recommendation engines perform several steps to mine the data. First, you begin with input data that represents the objects as well as their relationships.

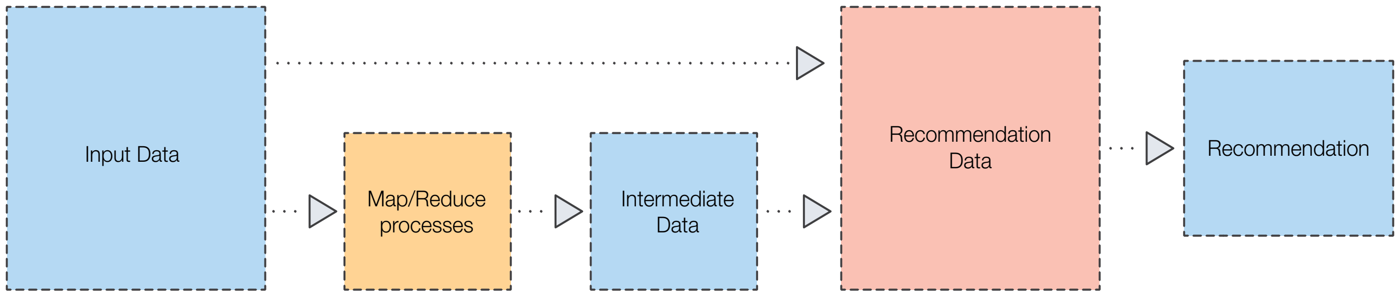

Figure 1: The flow of data into a recommendation engine

Consider the ratings users give to items. Using this input data, a recommendation engine computes a similarity between objects. Computing the similarity between objects can take time depending on the size of the data or the particular algorithm.

Deciding Which Algorithm to Use

The most important decision you'll need to make is what algorithm to apply to your data. The selection of the algorithm will depend on what you want to identify and what type of relationship is specified in your data. Some of the common approaches used for recommendations include:

Collaborative filtering:

This approach relies on the social interaction between users. The recommendations are based on rankings provided by other users.

Content-Based Filtering:

Recommendations made by content-based filters use the individual user's historical information to inform choices displayed. Browsing history is a major part of this, and provides a somewhat predictable solution for finding quality recommendations.

Clustering:

With this approach, the recommendation engine tries to build recommendations based on the similarities between either the users or the items themselves.

Categorization:

This approach automatically groups items together into categories using common attributes. In categorization, the computer attempts to classify all the items.

More About Collaborative Filtering

Of these algorithms, collaborative filtering is easy and a popular technique. It's easy because your customers do the important work, they drive the criteria of what you want to highlight. Collaborative filtering analyzes ratings from other users or items to make recommendations. There are two approaches within collaborative filtering, the main difference between them is the ability of each to scale as the number of users in the system grows.

- **User-based Recommendation:** This type of recommendation builds similarities between users by looking at the commonalities of the items rated by each user. For example, if the items are courses, two users could be considered very similar if they both took the same courses. In the other extreme, their similarity would be low if they did not take any similar course. To make recommendations, the algorithms rely on the ratings that similar users gave to those courses not taken by the user.
- **Item-based recommendation (like Amazon's "Customers Who Bought This" recommendations):** Item-based recommendation begins by looking at the items that are associated with the user. For each item associated with the user, the algorithm computes how similar it is to the other items in the collection to build the list of recommendations. To determine how likely the user is to appreciate a recommended item, the algorithm looks at the ratings that the user has given to the item and gives a weighted rating to each recommended item.

Below is a diagram of what a full recommendation pipeline looks like in production, with a green box around the portion of the pipeline that is powered by open code running on AWS:

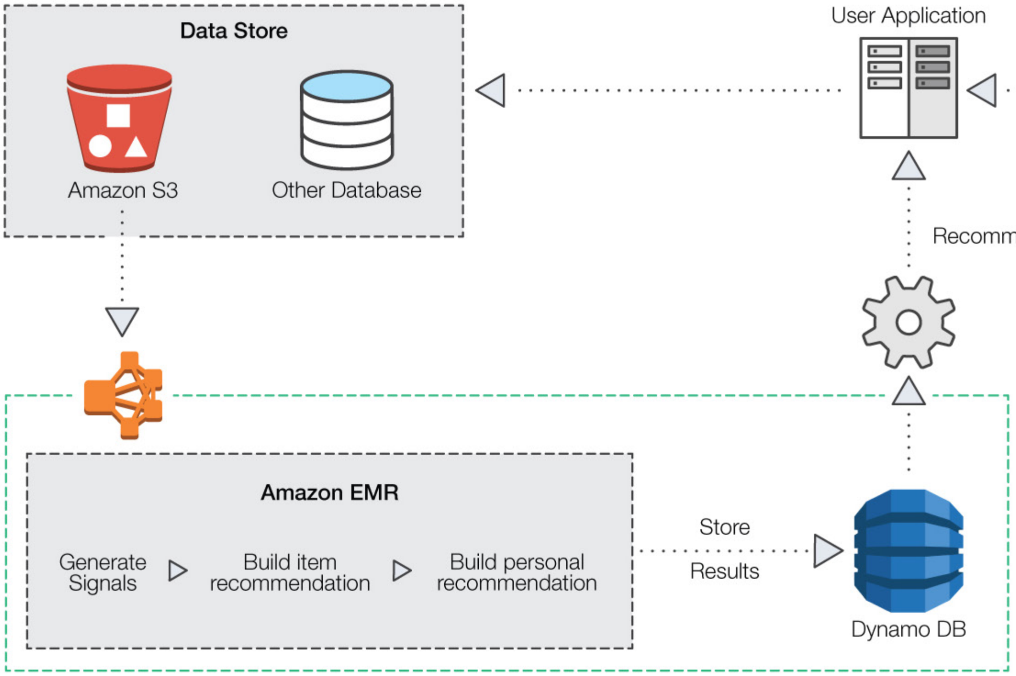

Figure 2: Recommendation pipeline; green portion is open code running on AWS

Where to Store Your Data

Businesses need to determine where to store their data. This data can contain the raw input data or the data with the similarities already computed by an offline process. If the source of your data is a large amount of raw data, you might need to mine it to get something to feed the recommendation engine. You can store your data sets in a file system, a distributed data source, or an in-memory database.

Conclusion

Once you're satisfied with the output from the recommendation engine, you can then put it in production. Recommendation services have become a standard expectation of web users in a wide variety of domains. When used correctly, recommendations can improve your business by delivering relevant content, increasing average order value, increasing items per order, converting shoppers to customers, and having visitors more actively engaged on your site.

Visit aws.amazon.com/mp/recommendations for more information.

Get Started with Bi-Data Analytics at AWS Marketplace

Find and deploy the solution you need in minutes

Save money with pay-as-you-go pricing

Scale globally across all AWS regions