ORACLE
Cloud Infrastructure
Data Science

Introduction to Recommendation Engines

A guide to algorithmically predicting what your customers want and when.

July 2020 | Version 1.01

TABLE OF CONTENTS

INTRODUCTION

Recommendation engines have become a popular solution for online retailers and streaming content companies looking to suggest products and media to users. Also known as recommender systems, these tools filter out less relevant information in order to predict how likely a user is to purchase an item or engage with certain videos or images, and suggest those things to the user.

The benefits of such systems are obvious: increased customer engagement, reduced churn, and an expanded reach, among other perks. And the associated boost to the bottom line is nothing to sneeze at. Amazon estimates that its recommendation system drives a 20 to 35 percent lift in sales[1], and Netflix values its system at US$1 billion per year[2].

The key to creating a powerful—and valuable—recommendation engine is to not treat it like a black box. Even a basic understanding of how these systems function will go a long way toward maximizing your investment, such as knowing what data you need to feed the system or what strategies to use when a new customer engages with it.

This white paper provides an introduction to the internal workings of recommendation engines to help you assess whether one is right for your business.

Recommendation engines can deliver the following benefits:

- Increased customer engagement
- Reduced churn
- Expanded reach
- Increased revenue

WHAT PROBLEMS ARE RECOMMENDATION ENGINES SOLVING?

The advent of streaming content and internet-based retail has brought about an age of product abundance. Traditional brick-and-mortar retailers can only stock the most popular items due to shelf-space constraints, but those that do business online don't face this limitation. Internet businesses can provide niche items that make up the "long tail" of the product lifecycle (Figure 1). Collectively, these niche items are worth a lot of money to those who market them strategically, giving companies a leg up on competitors with more standard inventory.

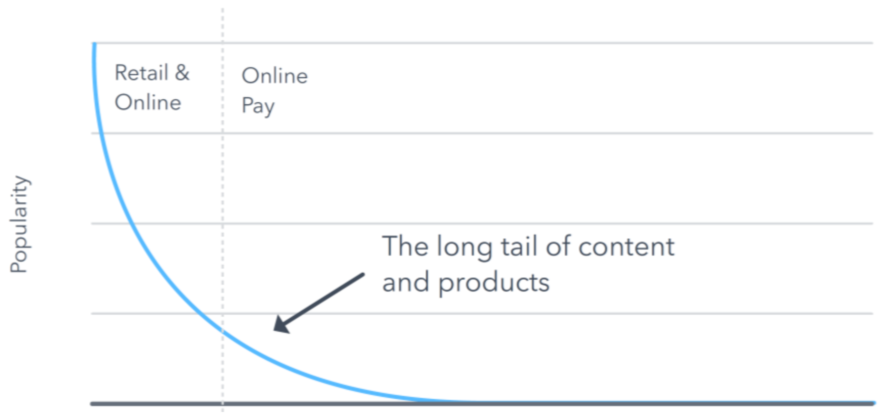

Figure 1. Products often decline in popularity over time, but internet business models enable consumers to find niche products later in their lifecycles when they have less availability.

However, getting niche products in front of the right users at the right time is not easy. This is precisely what recommender systems attempt to do by personalizing each user's experience. Recommendation engines filter out less relevant items and surface those that they predict will be more attractive to a customer. All it takes is the right information, an algorithm, and a little data science know-how.

WHAT DO YOU NEED TO BUILD A RECOMMENDATION ENGINE?

Recommender systems are very powerful, but they are not ideal for every business situation. There are several criteria that your business needs to satisfy before a full-blown system is worth the investment:

- You should be generating high-quality data at a high volume.
- You should ideally be generating multiple low-to-medium value transactions per customer.
- You should have sufficient data science and engineering resources to stand up and maintain your system.
- You should have information about your existing products and users to account for the problems that arise when a new product or user engages with your system.

Assuming you have satisfied these requirements, the next step is to make sure you have the right data on hand to build your system. This data required is similar to what you might use to make suggestions to someone you know in a face-to-face interaction. For example, the most basic engine will need a list of products and attributes associated with those products. (Data scientists would call these "items" and their "features"). For a more sophisticated engine, you will need a measure of each user's affinity for each product or item. This could be an explicit measure such as a rating or number of purchases, or an implicit measure such as browsing history or the time spent viewing a listing.

A recommendation engine is worth your investment if you are

- Generating large volumes of high-quality data.
- Generating multiple low-to-medium value transactions per customer.
- Employing sufficient data science and engineering resources to create and maintain a system.
- Collecting information about your existing products and users.

HOW DOES A RECOMMENDATION ENGINE WORK?

The two most common approaches to creating a recommendation engine are *content-based filtering* and *collaborative filtering*. If you've ever used Pandora's streaming music service, you have encountered a content-based system; if you've made a visit to Amazon or Netflix, you've experienced a collaborative filtering system. This is an oversimplification, of course. In many cases, companies use hybrid engines that effectively use the strengths—and offset the weaknesses—of each approach.

Content-based filtering:

Recommendations are based on data derived from each item.

Collaborative filtering:

Recommendations are based on a measure of a consumer's affinity for the item.

Just a note: In production settings, the mathematical techniques employed to create recommendation engines are often more sophisticated than what is described below. However, the end goal and intuition of these approaches remain much the same.

Content-based filtering

As the name suggests, content-based filtering uses data that are derived from each item itself. For an item of clothing, these could be the color, material, and designer of a piece of clothing. The list of features for each item is used to create its profile.

User preference profiles are generated by combining the profiles of items in a user's history. These two axes of information can then be leveraged in a user-item content-based filtering engine. The engine measures the mathematical distance between a user's profile and items that you would like to suggest, and uses that information to make recommendations. This distance can be quantified in many ways. One example is cosine similarity, which measures the angle between the user and item profiles to quantify their divergence from each other.

Cold-start issues are problems that arise when a new product is introduced to the system. A big advantage to the content-based filtering approach is that there are no cold-start issues for new items because item profiles don't depend on historical data. Compared to collaborative filtering, content-based filtering is more feasible to implement for smaller or newer businesses because it does not depend on a large, active user base to be truly effective.

However, a cold-start problem still exists for new users entering the system, as no concrete information about their preferences is available. Additionally, feature extraction and selection can sometimes be difficult and time consuming. Recommendations from a content-based filtering system may also lack diversity because the system is making suggestions based solely on one user's profile at a time, meaning items from novel categories are sometimes underrepresented.

Collaborative filtering

In systems using the collaborative filtering technique, information inherent to an item such as color, price, or material is not needed. Instead, a measure of each user's affinity for the items you are offering, perhaps in the form of ratings, is necessary.

These data can be summarized in a user-item matrix—essentially a giant table where each user is a row, each item is a column, and the cells are populated with ratings. You're almost guaranteed to have many more empty cells than filled ones; how you address that will depend on your situation. Then, you simply measure the distance between pairs of users—or use cosine similarity—to match a user with other users who are most similar to him or her. You can then predict how a user who hasn't encountered a particular item will feel about it, based on the ratings of similar users who have encountered it.

By leveraging inter-user information, collaborative filtering is better able to adapt to users with idiosyncratic tastes than content-based filtering. Collaborative systems can surface items that are novel in terms of that user's tastes, and subsequently, the diversity of recommendations is often higher than those of content-based systems.

However, a major issue with collaborative filtering is that it faces cold-start problems with both new users and items because it relies entirely on historical data. The difficult problem of accounting for variation in user behavior—some users may be easy raters while others are much tougher—must also be addressed. Lastly, the recommendations produced by such a system can be less interpretable than those from content-based filtering, as the reasoning behind a collaborative recommendation is not as clear as a content-based one to users.

HOW CAN I MEASURE MY SYSTEM'S PERFORMANCE?

There are many standard machine learning metrics that can be used to evaluate your recommendation system. For instance, to assess the accuracy of predicted values—such as ratings—you can use root-mean-square error (RMSE). If the ranking of recommended items is of interest, mean average precision (MAP) and normalized discounted cumulative gain (NDCG) are good options. After the engine has been deployed, you can look at whether a KPI of interest, such as the number of purchases, has changed between the pre- and post-implementation periods.

Although less easily quantified, there are additional metrics that are just as important to consider:

- **The diversity of the recommendations being made.** A recommender system that suggests obvious items such as Star Wars movies to the majority of users may need to be fine-tuned.
- **The relevance and usefulness of recommendations to your users.** Do users like getting the suggestions and find them helpful?
- **The overall impact of recommendations.** Recommender systems can be used to guide user behavior; if that is your goal, it is vital to study whether users are modifying their behavior in response to the information presented.

Machine Learning Metrics

These metrics can evaluate your recommendation system.

- **RMSE.** Root-Mean-Square Error assesses the accuracy of predicated values such as ratings.
- **MAP.** Mean Average Precision provides insight into the rankings of recommended items.
- **NDCG.** Normalized Discounted Cumulative Gain also evaluates the rankings of recommended items.

CONCLUSION

As with all major investments, there are many factors to examine before you start building a recommendation engine. The process of creating—and then maintaining—such a system is very expensive; not just in terms of financial cost, but also in technical resources. Recommendation engines must be retrained as new data is collected or their value and impact will diminish dramatically over time.

If implemented correctly, a recommendation engine can impact your business in a way that few algorithmic tools can match. Success hinges on aligning your system with your business goals, whether that's maximizing the value of your niche inventory or expanding the circulation of a content piece. All it takes is a little data science.

REFERENCES

1. Marshall, Matt. “Aggregate Knowledge Raises $5M from Kleiner, On a Roll.” Venturebeat.com, December 10, 2006.
2. McAlone, Nathan. “Why Netflix Thinks Its Personalized Recommendation Engine is worth $1 Billion per Year.” BusinessInsider.com. June 14, 2016.

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com/data-science.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com/datascience/ facebook.com/oracle twitter.com/oracledatasci

Introduction to Recommendation Engines
July, 2020