

f (<https://www.facebook.com/AnalyticsVidhya>)

t (<https://twitter.com/analyticsvidhya>)

g+ (<https://plus.google.com/+Analyticsvidhya/posts>)

in (<https://www.linkedin.com/groups/Analytics-Vidhya-Learn-everything-about-5057165>)



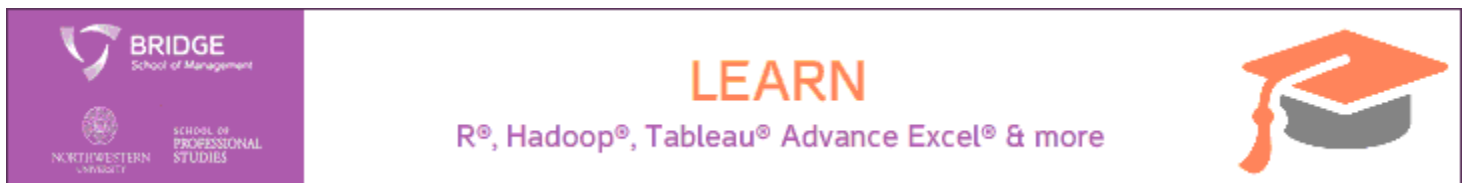
(<https://datahack.analyticsvidhya.com/contest/the-strategic-monk/>)

Home (<https://www.analyticsvidhya.com/>) > Business Analytics (<https://www.analyticsvidhya.com/blog/category/business-ana..>)

Understanding basics of Recommendation Engines (with case study)

BUSINESS ANALYTICS (<https://www.analyticsvidhya.com/blog/category/business-analytics/>)

ook.com/sharer.php?u=[https://www.analyticsvidhya.com/blog/2015/10/recommendation-20basics%20of%20Recommendation%20Engines%20\(with%20case%20study\)\)](https://www.analyticsvidhya.com/blog/2015/10/recommendation-20basics%20of%20Recommendation%20Engines%20(with%20case%20study))) **t** (<https://twitter.com/home?source=share&url=https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/>) **p** [https://www.analyticsvidhya.com/wp-content/uploads/2015/10/recommendation-engines-1-things2.jpg&description=Understanding%20basics%20of%20Recommendation%20Engines%20\(with%20case%20study\)\)](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/recommendation-engines-1-things2.jpg&description=Understanding%20basics%20of%20Recommendation%20Engines%20(with%20case%20study)))



(http://admissions.bridgesom.com/pba-new/?utm_source=AV&utm_medium=BannerInline&utm_campaign=AVBanner20August)

Introduction

Ever wondered, “what algorithm google uses to maximize its target ads revenue?”. What about the e-commerce websites which advocates you through options such as ‘people who bought this also bought this’. Or “How does Facebook automatically suggest us to tag friends in pictures”?

The answer is Recommendation Engines. With the growing amount of information on world wide web and with significant rise number of users, it becomes increasingly important for companies to search, map and provide them with the relevant chunk of information according to their preferences and tastes.

Companies nowadays are building smart and intelligent recommendation engines by studying the past behavior of their users. Hence providing them recommendations and choices of their interest in terms of “Relevant Job postings”, “Movies of Interest”, “Suggested Videos”, “Facebook friends that you may know” and “People who bought this also bought this” etc.



(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/few-things2.jpg>)

What are Recommendation Engines ?

Often termed as Recommender Systems, they are simple algorithms which aim to provide the most relevant and accurate items to the user by filtering useful stuff from of a huge pool of information base. Recommendation engines discovers data patterns in the data set by learning consumers choices and produces the outcomes that co-relates to their needs and interests.

Types of Recommendation Engine:

In this article, we will explain two types of recommendation algorithms that are also used by most of the tech giants like Google and Facebook in their advanced recommender system modules.

As a typical business problem,

Consider a scenario of an e-commerce website which sells thousands of smartphones. With growing number of customers every day, the task in hand is to showcase the best choices of smartphones to the users according to their tastes and preferences.

To understand how recommendation engine works, let's slice the data into a sample set of five smartphones with two major features "Battery and Display". The five smartphones have following properties:

- S1 has good battery life but poor display
- S2 has an amazing battery performance but very rough display
- S3's battery is one of the best but display lacks quality
- S4 & S5 are good in terms of display but poor in terms of battery performance.

Using these characteristics, we can create an **Item – Feature Matrix**. Value in the cell represents the rating of the smartphone feature out of 1.

Item – Feature Matrix

Smartphone	Battery	Display
S1	0.9	0.1
S2	1	0
S3	0.99	0.01
S4	0	1
S5	0.1	0.9

(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/F1.png>)

Our sample set also consist of four active users with their preferences.

- **Aman:** He prefers battery over display as an ideal smartphone feature.
- **Bob:** He likes a long lasting battery.
- **Chandan:** For Chandan, display should be decent, battery should be normal.

- **David:** For David, Display is extremely important but not the battery.

Using their interests, we can create a **User – Feature Matrix** as follows:

User	Battery	Display
Aman	0.9	0.1
Bob	0.8	0.2
Chandan	0.1	0.9
David	0.01	0.99

(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/F2.png>)

We have two matrices: *Item – Feature* and *User – Feature*. We can create the recommendation of smartphones for our users using following algorithms:

Content Based Recommendations

Content based systems, recommends item based on a similarity comparison between the content of the items and a user's profile. The feature of items are mapped with feature of users in order to obtain user – item similarity. The top matched pairs are given as recommendations, as demonstrated below: Representing every user by a feature vector:

	User	Feature Vector
U ₁	Aman	[0.9 0.1]
U ₂	Bob	[0.8 0.2]
U ₃	Chandan	[0.1 0.9]
U ₄	David	[0.01 0.99]

(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/F3.png>)

Also, every item representation as a feature vector:

Smartphone	Feature Vector
S ₁	[0.9 0.1]
S ₂	[1 0]

(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/F4.png>)

and so on...

Content Based **Item – User Mapping Recommendations** are given by the equation:

$$\text{MAX} (U_{(j)}^T \cdot I_{(i)})$$

$$i, j \rightarrow n, m$$

For User U₁ (Aman), Smartphone recommendation is:

$$\text{MAX}(U_1^T S_1, U_1^T S_2, U_1^T S_3, U_1^T S_4, U_1^T S_5)$$

$$\text{MAX}([0.9 \ 0.1]^T [0.9 \ 0.1], [0.9 \ 0.1]^T [1 \ 0], [0.9 \ 0.1]^T [0.99 \ 0.01], [0.9 \ 0.1]^T [0.1 \ 0.9], [0.9 \ 0.1]^T [0.01 \ 0.99])$$

$$\text{MAX}(0.82, 0.9, 0.89, 0.18, 0.10)$$

$$= S_2(0.9), S_3(0.89) \ \& \ S_1(0.82)$$

Smartphones S₂, S₃ and S₁ has the highest recommendation scores, Hence S₂, S₃ and S₁ are recommended to Aman.

Collaborative Filtering

Content-based recommendation lacks in detecting inter dependencies or complex behaviors. For example: People might like smartphones with Good Display, only if it has retina display and wouldn't otherwise.

Collaborative Filtering algorithm considers "User Behaviour" for recommending items. They exploit behaviour of other users and items in terms of transaction history, ratings, selection and purchase information. Other users behaviour and preferences over the items are used to recommend items to the new users. In this case, features of the items are not known.

We have a similar **User – Feature Matrix** as content based:

User – Feature Matrix

User	Battery	Display	Feature Vector
Aman	0.9	0.1	[0.9 0.1]
Bob	0.8	0.2	[0.8 0.2]
Chandan	0.1	0.9	[0.1 0.9]
David	0.3	0.7	[0.01 0.99]

(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/F8.png>)

This time we don't know features of the items but we have user behaviour. i.e. How the Users brought/rated the existing items.

User- Behaviour Matrix

Smartphone	Aman	Bob	Chandan	David
S1	5	4.5	?	?
S2	5	?	0.5	?
S3	?	4	0.5	?
S4	?	?	5	4
S5	?	?	5	4.5

(<https://www.analyticsvidhya.com/wp-content/uploads/2015/10/F7.png>)

where values of the behaviour matrix can be described as:

$$B_{i,j} = \begin{cases} r, & \text{if } U_j \text{ has given "r" rating to a } S_i \\ ?, & \text{if no rating is given} \end{cases}$$

This user behavior matrix can be used to derive unknown features of the most liked items. Lets try to derive features of S1 using this behavior matrix.

S1 is rated 5 by U1

S1 is rated 4.5 by U2

S1 rating by U3 & U4 are not known

Using this information Feature Vector of S1 can be assumed as:

$$S1 : [x_1 \ x_2]$$

and the equations are:

$$U_1^T S_1 = 5$$

$$U_2^T S_1 = 4.5$$

$$[0.9 \ 0.1]^T [x_1 \ x_2] = 5$$

$$[0.8 \ 0.2]^T [x_1 \ x_2] = 4.5$$

$$0.9 * x_1 + 0.1 * x_2 = 5$$

$$0.8 * x_1 + 0.2 * x_2 = 4.5$$

solving these equations, gives $x_1 = 5.5$ and $x_2 = 0.5$

$$S1 = [5.5 \quad 0.5]$$

Similarly,

$$S2 = [5.5 \quad 0]$$

$$S3 = [5 \quad 0]$$

$$S4 = [0.5 \quad 5.5]$$

$$S5 = [2.7 \quad 5.25]$$

Now all the feature vectors are known. Hence the recommendations will be mappings of User Feature Vectors and Item Feature Vectors. Thus for Aman, based on his preferences and behaviours, recommendation will be:

$$\text{MAX}(U_1^T S_1, U_1^T S_2, U_1^T S_3, U_1^T S_4, U_1^T S_5)$$

$$\text{MAX}([0.9 \quad 0.1]^T [5.5 \quad 0.5], [0.9 \quad 0.1]^T [5.5 \quad 0], [0.9 \quad 0.1]^T [5 \quad 0], [0.9 \quad 0.1]^T [0.5 \quad 5.5], [0.9 \quad 0.1]^T [2.7 \quad 5.25])$$

$$\text{MAX}(5, 4.99, 4.95, 1, 2.9)$$

$$> S1, S2 \text{ and } S3$$

which comes out to be S1, S2 and S3 again. Since S1 and S2 are already rated by Aman, So we will recommend him a new smartphone S3.

In the above example where we assumed that there are two primary features of S1 as governed by the users who rated it. In real case, we end up with more number of features than this. For example, if we had data for all the N number of users who rated S1, then feature vector look like:

S1: [x1 x2 x3 x4 x5 ...]

End notes

In this article, we learnt about two types of Recommendation Engines: Content based recommendations and Collaborative Recommendations. There exists more advanced techniques like ALS : Alternating Least Square Recommendations and Hybrid Recommendation Engines. The Recommendation Engines have become an important need with the growing information space. Did you find this article useful ? Have you also worked on recommender systems? Share your opinions / views in the comments section below.

If you like what you just read & want to continue your analytics learning, subscribe to our emails (<http://feedburner.google.com/fb/a/mailverify?uri=analyticsvidhya>), follow us on twitter (<http://twitter.com/analyticsvidhya>) or like our facebook page (<http://facebook.com/analyticsvidhya>).

Share this:


 (<https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/?share=linkedin&nb=1>) 159

 (<https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/?share=facebook&nb=1>) 2

 (<https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/?share=google-plus-1&nb=1>)

 (<https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/?share=twitter&nb=1>)

 (<https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/?share=pocket&nb=1>)

 (<https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/?share=reddit&nb=1>)

RELATED
