



## Data Science Case Study

# Recommender Engine Design

*Mosaic designed & deployed a custom recommendation engine for a social media app.*



### Industry

Social Media



### Use Case

Recommender Engine  
Development



### Techniques

Machine Learning,  
Collaborative and  
content-based filtering



### Outcome

Recommender delivered  
that can model the  
social connections of 1M  
users every six hours.

## SO WE NEED A RECOMMENDATION ENGINE<sup>1</sup>

The customer wanted to develop a social networking site which connects users to like-minded businesses and activities. Understanding the current options in the marketplace, the customer believed the addition of a recommender engine would give their application significant competitive advantage.

However, with a launch date six months out, it would be hard to create a recommender engine because there was no customer data to analyze.

How could a data science company provide a solution with no data?

## ALGORITHM DESIGN

With no real data on which to develop and test the recommender, the recommender models would need to be able to dynamically learn as data entered the system and to be easily tuned by analysts to better model observed patterns.

Mosaic determined that a hybrid *collaborative* and *content-based filtering* machine learning model would provide the necessary performance and flexibility. Content-based filtering makes use of known data about site users and “items” – businesses, advertisements, in-site product offers, etc. – to identify patterns in user preferences. For example, users with a specific set of demographic traits may tend to accept offers from businesses of a given type. These models are intuitive and can quickly begin generating relevant recommendations even for users that are new to the site.

Mosaic’s patent-pending machine learning model blended the three model types – content-based filtering and both user-based and item-based collaborative filtering – for the recommender engine design and is able to leverage the strengths of each approach while maintaining model stability. This allows the recommender to gradually adjust its recommendations as a user moves from being newly registered with only demographic data available to being an experienced user with a long history of interactions on the site. Mosaic built in a number of “dials” allowing site administrators and analysts to tune various model parameters: the relative weighting of different user behaviors in determining preferences, how the content-based and collaborative filtering models are blended as more user data becomes available, how user and item similarity are determined, etc.



## RECOMMENDER DEPLOYMENT

Technical requirements from our customer dictated that our solution should be able to model the social connections of 1 million users every six hours. We provisioned a cluster of four cloud-based Linux servers from RackSpace to run our model.

The model was implemented in a version of Mahout that we optimized to leverage specific aspects of our model and to meet the customer's specific performance objectives. We implemented an improved disk-caching scheme and maximized in-memory processing for improved performance. We also prepared our data to support sequential iteration of data elements for critical calculations rather than performing map lookups.

We were able to leverage the geographic nature of the problem based on the knowledge that user interactions were only computed within discrete geographic bounds. This allowed us to segment data into relatively small sets and distribute the data across processing nodes according to these geographic boundaries, thereby limited the amount of data processed by any single node in any one execution of our model.

## Endnotes

1. <https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine>

