

# Faber Ventures Challenge

Welcome to our challenge.

This README will guide you on how to participate in this challenge.

Please understand that this challenge is not decisive. There are no right and wrong answers.

## Challenge

---

### Build a recommendation engine.

Using the supplied dataset build a a simple recommendation engine that given a user ID ( `userId` ) returns a collection (list) of recommended movies.

The recommendation engine can be written in any language (ruby, python, php, javascript, bash, etc) and built using any technology/framework but you are required to state why you chosen that language and the technology stack.

Bonus points who builds the solution integrated with a web interface.

Please comment your code.

## Resources

---

### Dataset

This dataset consists of movie reviews from Amazon. The data span a period of more than 10 years, including all ~8 million reviews up to October 2012. Reviews include product and user information, ratings, and a plaintext review.

Dataset: <http://snap.stanford.edu/data/web-Movies.html>

Each entry of the dataset looks like this:

```
product/productId: B00006HAXW
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than t
he
1st ones. Remember once these performers are gone, we'll never get to see them again.
Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOV
E
this DVD !!
```

## Where

- product/productId: asin, e.g. amazon.com/dp/B00006HAXW
- review/userId: id of the user, e.g. A1RSDE90N6RSZF
- review/profileName: name of the user
- review/helpfulness: fraction of users who found the review helpful
- review/score: rating of the product
- review/time: time of the review (unix time)
- review/summary: review summary
- review/text: text of the review

## Notice

---

If for some reason you can't build a solution (be it for not having time to do it or programming knowledge) please submit a text version in your own words describing how you would implement the algorithm or how it would work.

## Delivering the solution

---

The solution should be published on [Github](#). Once completed the work send the repository URL to

`sofia@faber.vc` with the subject

`Faber Ventures Challenge 2017 - <firstname> <lastname>` .

Don't forget to include instructions on how to setup, build and run your project.

Remember to commit often so that we can see your progress while developing the solution.

