Sign up for an AWS account. Start up an EMR cluster Get the MovieLens data

```
wget http://files.grouplens.org/datasets/movielens/ml-1m.zip
unzip ml-1m.zip
```

Convert ratings.dat, trade “::” for “,”, and take only the first three columns:

```
cat ml-1m/ratings.dat | sed 's/::/,/g' | cut -f1-3 -d, > ratings.csv
```

Put ratings file into HDFS:

```
hadoop fs -put ratings.csv /ratings.csv
```

Run the recommender job:

```
mahout recommenditembased --input /ratings.csv --output recommendations
--numRecommendations 10 --outputPathForSimilarityMatrix similarity-matrix
--similarityClassname SIMILARITY_COSINE
```

Look for the results in the part-files containing the recommendations:

```
hadoop fs -ls recommendations
 hadoop fs -cat recommendations/part-r-00000 | head
```

You should see a lookup file that looks something like this (your recommendations will be different since they are all 5.0-valued and we are only picking ten):

User ID (Movie ID : Recommendation Strength) Tuples 35 [2067:5.0, 17:5.0, 1041:5.0, 2068:5.0, 2087:5.0, 1036:5.0, 900:5.0, 1:5.0, 2081:5.0, 3135:5.0] 70 [1682:5.0, 551:5.0, 1676:5.0, 1678:5.0, 2797:5.0, 17:5.0, 1:5.0, 1673:5.0, 2791:5.0, 2804:5.0] 105 [21:5.0, 3147:5.0, 6:5.0, 1019:5.0, 2100:5.0, 2105:5.0, 50:5.0, 1:5.0, 10:5.0, 32:5.0] 140 [3134:5.0, 1066:5.0, 2080:5.0, 1028:5.0, 21:5.0, 2100:5.0, 318:5.0, 1:5.0, 1035:5.0, 28:5.0] 175 [1916:5.0, 1921:5.0, 1912:5.0, 1914:5.0, 10:5.0, 11:5.0, 1200:5.0, 2:5.0, 6:5.0, 16:5.0] 210 [19:5.0, 22:5.0, 2:5.0, 16:5.0, 20:5.0, 21:5.0, 50:5.0, 1:5.0, 6:5.0, 25:5.0] 245 [2797:5.0, 3359:5.0, 1674:5.0, 2791:5.0, 1127:5.0, 1129:5.0, 356:5.0, 1:5.0, 1676:5.0, 3361:5.0] 280 [562:5.0, 1127:5.0, 1673:5.0, 1663:5.0, 551:5.0, 2797:5.0, 223:5.0, 1:5.0, 1674:5.0, 2243:5.0] Where the first number is a user id, and the key-value pairs inside the brackets are movie-id:recommendation-strength tuples.

The recommendation strengths are at a hundred percent, or 5.0 in this case, and should work to finesse the results. This probably indicates that there are many more than ten “perfect five” recommendations for most people, so you might calculate more than the top ten or pull from deeper in the ranking to surface less-popular items.

Building a Service Next, we’ll use this lookup file in a simple web service that returns movie recommendations for any given user.

Get Twisted, and Klein and Redis modules for Python.

```
sudo pip3 install twisted
 sudo pip3 install klein
 sudo pip3 install redis
```

Install Redis and start up the server.

```
wget http://download.redis.io/releases/redis-2.8.7.tar.gz
 tar xzf redis-2.8.7.tar.gz
 cd redis-2.8.7
 make
 ./src/redis-server &
```

Build a web service that pulls the recommendations into Redis and responds to queries. Put the following into a file, e.g., “hello.py”

```
from klein import run, route
import redis
import os

# Start up a Redis instance
r = redis.StrictRedis(host='localhost', port=6379, db=0)

# Pull out all the recommendations from HDFS
p = os.popen("hadoop fs -cat recommendations/part*")

# Load the recommendations into Redis
for i in p:

  # Split recommendations into key of user id
  # and value of recommendations
  # E.g., 35^I[2067:5.0,17:5.0,1041:5.0,2068:5.0,2087:5.0,
  #       1036:5.0,900:5.0,1:5.0,081:5.0,3135:5.0]$
  k,v = i.split('t')

  # Put key, value into Redis
```

```
  r.set(k,v)

# Establish an endpoint that takes in user id in the path
@route('/<string:id>')

def recs(request, id):
  # Get recommendations for this user
  v = r.get(id)
  return 'The recommendations for user '+id+' are '+v.decode()


# Make a default endpoint
@route('/')

def home(request):
  return 'Please add a user id to the URL, e.g. http://localhost:8083/1234n'

# Start up a listener on port 8083
run("localhost", 8083)
```

Start the web service.

```
twistd -noy hello.py &
```

Test the web service with user id “35”:

```
curl localhost:8083/35
```

You should see a response like this (again, your recommendations will differ): The recommendations for user 35 are [7:5.0,2088:5.0,2080:5.0,1043:5.0,3107:5.0,2087:5.0,2078:5.0,3108:5.0,1042:5.0,1028:5.0] When you’re finished, don’t forget to shut down the cluster