CS5229 Mahout Assignment

M.R.G.Vijithasena-209387N

Github Repo

https://github.com/rasikavijithasena/mahoutRecommendation

Steps that followed in assignment

1. Create an EMR cluster with mahout.
2. Get movieLens data and unzip zip file
   ```
   wget http://files.grouplens.org/datasets/movielens/ml-1m.zip
   unzip ml-1m.zip
   ```
3. Convert :: into , in ratings.dat file and select first 3 columns.
   ```
   cat ml-1m/ratings.dat | sed 's/::/,/g' | cut -f1-3 -d, > ratings.csv
   ```
4. Put ratings.csv into HDFS
   ```
   hadoop fs -put ratings.csv /ratings.csv
   ```
5. Run the recommender job
   ```
   mahout recommenditembased --input /ratings.csv --output
   recommendations --numRecommendations 10
   --outputPathForSimilarityMatrix similarity-matrix
   --similarityClassname SIMILARITY_COSINE
   ```

6. View the results in part-files
   ```
   hadoop fs -ls recommendations
   hadoop fs -cat recommendations/part-r-00000 | head
   ```

   Building a web service

   Web service which give movie recommendations for any user
7. Get install twisted, klein and redis modules to python
   ```
   sudo easy_install twisted
   sudo easy_install klein
   sudo easy_install redis
   ```
8. Install Redis and startup 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 &
   ```
9. Create hello.py file
10. Start the web service
    ```
    twistd -noy hello.py &
    ```

11. Test the web service with id 37

    ```
    curl localhost:8080/37
    ```

12. Then get the recommendation as following.

    ```
    [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]
    ```

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
  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

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

def home(request):
  return 'Please add a user id to the URL, e.g.
http://localhost:8081/1234n'

# Start up a listener on port 8081
run("localhost", 8081)
```