**JSFIDDLE**

HTML

<div data-role="page" id="page1">

<div data-role= "header">

<h1>Book rating </h1>

</div>

<div data-role="content"><b> recommended books</b><input type="text" id="from">

<a href="#bookID"><button>bookID</button></a><br>

<div data-role="content"><b> Rating </b><br>

<div id="write"></div>

</div>

</div>

**JAVASCRIPT**

$.getJSON("http://134.193.136.127:8983/solr/collection1\_shard1\_replica1/select?q=\*&wt=json&json.wrf=?&indent=true", function(result){

document.getElementById('write').innerHTML= result.response.docs[i].title[j] ;

});

{

for( i=276725;i<276736;i++)

{

for(j=276725;j<276748;j++)

result.response.docs[i+2].title[j]

}

**CSS**

$('.ui-bar-b').css('background-image', '-moz-linear-gradient(top,

#00009d,

#00578e);');

**Recommendation algorithm**

package com.unresyst;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.List;

import java.io.IOException;

import org.apache.commons.cli2.OptionException;

import org.apache.mahout.cf.taste.common.TasteException;

import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;

import org.apache.mahout.cf.taste.impl.recommender.CachingRecommender;

import org.apache.mahout.cf.taste.impl.recommender.slopeone.SlopeOneRecommender;

import org.apache.mahout.cf.taste.model.DataModel;

import org.apache.mahout.cf.taste.recommender.RecommendedItem;

import org.apache.mahout.cf.taste.impl.common.LongPrimitiveIterator;

public class UnresystBoolRecommend {

public static void main(String... args) throws FileNotFoundException, TasteException, IOException, OptionException {

// create data source (model) - from the csv file

File ratingsFile = new File("datasets/books.csv");

DataModel model = new FileDataModel(ratingsFile);

// create a simple recommender on our data

CachingRecommender cachingRecommender = new CachingRecommender(new SlopeOneRecommender(model));

// for all users

for (LongPrimitiveIterator it = model.getUserIDs(); it.hasNext();){

long userId = it.nextLong();

// get the recommendations for the user

List<RecommendedItem> recommendations = cachingRecommender.recommend(userId, 100000);

// if empty write something

if (recommendations.size() == 0){

System.out.print("User ");

System.out.print(userId);

System.out.println(": no recommendations");

}

// print the list of recommendations for each

for (RecommendedItem recommendedItem : recommendations) {

System.out.print("User ");

System.out.print(userId);

System.out.print(": ");

System.out.println(recommendedItem);

}

}

}

}