**Program Code**

**Java code for clustering:**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

public class Main {

public static void main(String[] args) throws IOException {

int temp=0;

Map<String, String> map\_genre = new HashMap<>();

Map<String, String> map\_nation = new HashMap<>();

BufferedReader reader = new BufferedReader(new FileReader("/home/cloudera/Desktop/movies.dat"));

String line = null;

while ((line = reader.readLine()) != null) {

temp=0;

List<String> test= new ArrayList<String>();

String[] tokens=line.split("::",4 );

String genre=tokens[3];

String movie=tokens[2];

String county=tokens[1];

for(int i=0;i<genre.length();i++)

{

if(genre.charAt(i)=='|')

{

if(!test.contains(genre.substring(temp, i))) {

test.add(genre.substring(temp, i));

}

temp=i+1;

}

}

if(!test.contains(genre.substring(temp, genre.length()))) {

test.add(genre.substring(temp, genre.length()));

}

Iterator<String> iterator = test.iterator();

while (iterator.hasNext()) {

String category=iterator.next();

if(category.compareTo("")==0)

category="Other";

if(!map\_genre.isEmpty() && !map\_nation.isEmpty())

{

if(map\_genre.containsKey(category)&&map\_nation.containsKey(category))

{

String content=map\_genre.get(category)+","+movie;

String country=map\_nation.get(category)+","+county;

map\_genre.put(category, content);

map\_nation.put(category, country);

}

else

{

map\_genre.put(category, movie);

map\_nation.put(category, county);

}

}

else

{

map\_genre.put(category, movie);

map\_nation.put(category, county);

}

}

}

BufferedWriter reader1 = new BufferedWriter(new FileWriter("/home/cloudera/Desktop/output.txt"));

System.out.println(map\_genre.size()+"\n");

Iterator<Entry<String, String>> entries = map\_genre.entrySet().iterator();

Iterator<Entry<String, String>> entries1 = map\_nation.entrySet().iterator();

while (entries.hasNext()&&entries1.hasNext()) {

Map.Entry<String, String> entry = entries.next();

Map.Entry<String, String> entry1 = entries1.next();

String key=entry.getKey();

String desh=entry1.getValue();

String cinema=entry.getValue();

String [] key\_list=key.split(",");

String [] desh\_list=desh.split(",");

String [] cinema\_list=cinema.split(",");

String entireline2="NULL";

String oldkey="";

String entireline1="NULL";

/\* for (int i=0; i<desh\_list.length && i<cinema\_list.length;i++)

{

/\*String entireline="";

if (i ==0){

entireline += "{\"key \": \n {\"" + key\_list[i] + "\": [ \n";

oldkey=key\_list[i];

}

else if(key\_list[i]!=oldkey)

{

entireline += "]\n } \n{\"" + key\_list[i] + " \": \n [ \n";

}

entireline += "{\"country\": \"" + desh\_list[i] + "\",\"movie\"" + cinema\_list[i] + "}," + "\n";

reader1.write(entireline);

oldkey=key\_list[i];

}

\*/

String entireline="[{\"id\":\"" + entry.getKey() + "\"," +"\"title\":\"" + entry1.getValue() + "," + entry.getValue() + "\"}," + "\n";

reader1.write(entireline);}

reader1.close();reader.close();

}

}

**Java Code for recommending:**

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/movie.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, 10);

// 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);

}

}

}

}

**Program code for webapp:**

<meta charset="1244">

<title>Revive</title>

<body>

<p> <h1>

<centre>Watch While Eat...</centre>

</h1></p>

<form id="Country details" name="Country Details">

<label>Restaurants you wish: </label>

<select id="nation">

<option value="select one from below">Select one option</option>

<option value="Indian">Indian</option>

<option value="Chinese">Chinese</option>

<option value="Korean">Korean</option>

<option value="Japanese">Japanese</option>

<option value="American">American</option>

<option value="Mongolian">Mongolian</option>

<option value="Russian">Russian</option>

<option value="French">French</option>

</select>

<button id="Recommend">Recommend</button>

<br />

</form>

<p>Movie Recommendations:</p>

<div id="movies"></div>

<p>Restaurant recommendations for you:</p>

<div id="restaurants"></div>

</body>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js">t>

<script>

function on\_data(data) {

$('#results1').empty();

var users = data.response.docs;

$.each(users, function(id,title) {

var endlist = 'Recommended Movies'+users;

$('#results1').prepend('<div>' + endlist + '</div>');

}

function on\_search() {

var query = $('#Genre').val();

if (query.length == 0) {

return;

}

var url='http://134.193.136.127:8983/solr/collection1\_shard1\_replica1/select?q=id%3Aquery&wt=json&indent=true';

$.getJSON(url);

}

function on\_ready() {

$('#search').click(on\_search);

$('body').keypress(function(kp) {

if (kp.keyCode == '13') {

on\_search();

}

});

}

$(document).ready(on\_ready);