Name of the student:	Tanmay Prashant Rane	**Roll No.**	8031
Practical Number:	9	**Date of Practical:**	
Relevant CO's	**At the end of the course students will be able to apply Big data analytics in real life applications.**		
Sign here to indicate that you have read all the relevant material provided before attempting this practical		**Sign:**	

Practical grading using Rubrics

Indicator	Very Poor	Poor	Average	Good	Excellent
Timeline (2)	More than a session late (0)	NA	NA	NA	Early or on time (2)
Code design (2)	N/A	Very poor code design with no comments and indentation(0.5)	Poor code design with very comments and indentation (1)	Design with good coding standards (1.5)	Accurate design with better coding satndards (2)
Performance (4)	Unable to perform the experiment (0)	Able to partially perform the experiment (1)	Able to perform the experiment for certain use cases (2)	Able to perform the experiment considering most of the use cases (3)	Able to perform the experiment considering all use cases (4)
Postlab (2)	Incorrect answer(0)	N/A	Partially correct answer (1)	N/A	Fully correct answer (2)

Total Marks (10)	Sign of instructor with date

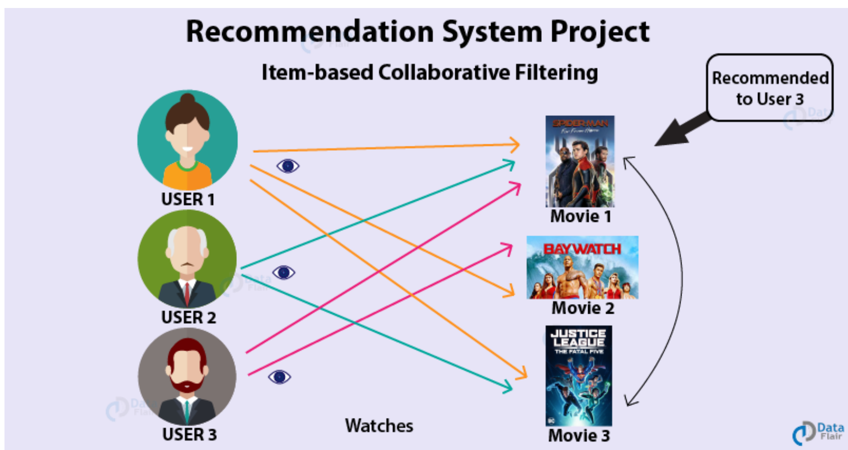
Practical

Course title: Big Data Analytics
Course term: 2019-2020
Instructor name: Saurabh Kulkarni

Problem Statement: To demonstrate use of recommendation system for movie rating prediction

Theory:

A **recommendation system** provides suggestions to the users through a filtering process that is based on user preferences and browsing history. The information about the user is taken as an input. The information is taken from the input that is in the form of browsing data. This information reflects the prior usage of the product as well as the assigned ratings. A recommendation system is a platform that provides its users with various contents based on their preferences and likings. A recommendation system takes the information about the user as an input. The recommendation system is an implementation of the machine learning algorithms.



A recommendation system also finds a similarity between the different products. For example, Netflix Recommendation System provides you with the recommendations of the movies that are similar to the ones that have been watched in the past. Furthermore, there is a collaborative content filtering that provides you with the recommendations in respect with the other users who might have a similar viewing history or preferences. There are two types of recommendation systems – Content-Based Recommendation System and Collaborative Filtering Recommendation.

Collaborative filtering uses the known behaviour of a group of users to make recommendations for the others. For instance, you can predict whether a particular user will like item A using the data of other users who have similar preferences about items. To apply collaborative filtering effectively, huge amount of data is required.

The second common approach, content-based filtering makes use of the comparison between items and the past preferences of a particular user. That is, the items which have similar properties to the ones that the user liked or checked previously are likely to be recommended.

Intercalarily, a recommendation system which uses both techniques together can be more suitable. Such systems are called Hybrid Recommendation Systems.

Using the correlation we can:

* For every pair of movies A and B, find all the people who rated both A and B.

* Use these ratings to form a Movie X vector and a Movie Y vector.

* Calculate the correlation between those two vectors

* When someone watches a movie, you can recommend the movies most correlated with it

Code:

Write R code for recommendation system for given input

code for the problem:

```
library(lsa)
critics = read.csv("/media/tanmay/Data/SEM-8/BDA/EXP9/Movieratings.csv")
#calculate the euclidian distance
#EUD = dist(critics[,2:7])
#cosine similarity calculation
x  = critics[,2:7]
x[is.na(x)] = 0
user_sim = cosine(as.matrix(t(x))) #user similarity
#Recommending items
#for Toby

#create weightge matrix
weight_mat = user_sim[,7]*critics[,2:7]

rec_itm_for_user = function(userNo)
{
  #calcualte column wise sum
  col_sums= list()
  rat_user = critics[userNo,2:ncol(critics)]
  x=1
  tot = list()
  z=1

  for(i in 1:ncol(rat_user)){
    if(is.na(rat_user[1,i]))
    {

      col_sums[x] = sum(weight_mat[,i],na.rm=TRUE)
      x=x+1

      temp = as.data.frame(weight_mat[,i])

      sum_temp=0

      for(j in 1:nrow(temp)){
        if(!is.na(temp[j,1])){
          sum_temp = sum_temp+user_sim[j,ncol(rat_user)]
        }

      }
      tot[z] = sum_temp
      z=z+1
    }


  }
  z=NULL

  z=1
  for(i in 1:ncol(rat_user)){
    if(is.na(rat_user[1,i]))
    {
      rat_user[1,i] = col_sums[[z]]/tot[[z]]
      z=z+1
    }

  }

  return(rat_user)
}


#to get N recommendations:
rec_itm_for_user(6) #first person recommendations
```

PostLab:
Explain Content based recommendation systems

Answer for postlab question

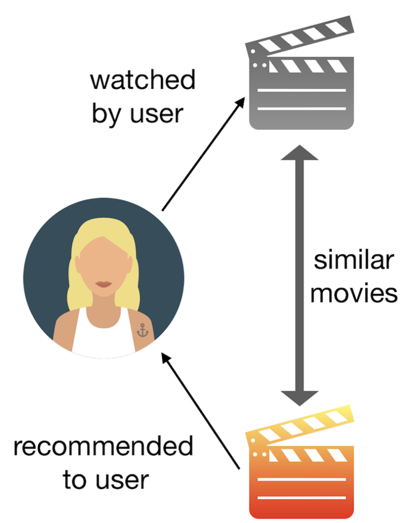
Content-Based systems focus on properties of items. Similarity of items is determined by measuring the similarity in their properties.

Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user.

Several issues have to be considered when implementing a content-based filtering system. First, terms can either be assigned automatically or manually. When terms are assigned automatically a method has to be chosen that can extract these terms from items. Second, the terms have to be represented such that both the user profile and the items can be compared in a meaningful way. Third, a learning algorithm has to be chosen that is able to learn the user profile based on seen items and can make recommendations based on this user profile.

The information source that content-based filtering systems are mostly used with are text documents. A standard approach for term parsing selects single words from documents. The vector space model and latent semantic indexing are two methods that use these terms to represent documents as vectors in a multi dimensional space.

Relevance feedback, genetic algorithms, neural networks, and the Bayesian classifier are among the learning techniques for learning a user profile. The vector space model and latent semantic indexing can both be used by these learning methods to represent documents. Some of the learning methods also represent the user profile as one or more vectors in the same multi dimensional space which makes it easy to compare documents and profiles. Other learning methods such as the Bayesian classifier and neural networks do not use this space but represent the user profile in their own way.



Explain Collaborative filtering systems

Answer for postlab question

Collaborative Filtering System:

This type of filter is based on users' rates, and it will recommend us movies that we haven't watched yet, but users similar to us have, and like. To determine whether two users are similar or not, this filter considers the movies both of them watched and how they rated them. By looking at the items in common, this type of algorithm will basically predict the rate of a movie for a user who hasn't watched it yet, based on the similar users' rates.

n order to work accurately, this type of filter needs rates, and not all users rate products constantly. Some of them barely or never rate anything! Another characteristic of this method is the diversity in the recommendations, that can be good or bad, depending on the case.

For example, let's say user A likes dystopian movies and dark comedy a lot. User B also enjoys dystopian movies but never watched dark comedy. The collaborative filter will recommend dark comedy shows to user B, based on the common taste that the two users have for dystopian movies. This scenario can go two ways: either user B finds out that he/she likes dark comedy a lot, and in that case, great, a lot of new things to watch on his/her list! Or, user B really enjoys a lighter comedy style, and in that case the recommendation has not been successful.