Ministry of Education and Science of Ukraine

National Technical University of Ukraine

«Kyiv Polytechnic Institute. Igor Sikorsky »

Faculty of Informatics and Computer Technologies

Department of Computer Engineering

LAB № 2

from the discipline "Theory of Algorithms"

on the topic «Decomposition method. Search for inversions»

PERFORMED BY:

1st year student

group ІП-93

Zavalniuk M.E.

The credit - 9312

Variant – 12

CHECKED:

Associate Professor of OT

c.t.s.,s.r.

Antoniuk А.І.

Kiev - 2020

**TASK**

**Goal:**

using the decomposition method to develop an algorithm and find the number of degrees of liking (number of inversions).

**Option task:**

Use a decomposition method to develop an algorithm that will solve the following problem.

Incoming data. Matrix D of natural numbers of dimension uxm, where u are the number of users, m - number of films. Each element of the matrix D [i, j] indicates the position of the movie j in the list of user preferences i. Another input is the x number of the user, which will be compared with the numbers of all other users.

Output data. List of ascending second element of pairs (i, c), where i is the user number, c is a number that indicates the degree of liking of users x and c (number of inversions).

**CODE**

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib**.**pyplot **as** plt

data **=** pd**.**io**.**parsers**.**read\_csv**(**'./ratings.dat'**,**

names**=[**'user\_id'**,** 'movie\_id'**,** 'rating'**,** 'time'**],**

engine**=**'python'**,** delimiter**=**'::'**)**

movie\_data **=** pd**.**io**.**parsers**.**read\_csv**(**'./movies.dat'**,**

names**=[**'movie\_id'**,** 'title'**,** 'genre'**],**

engine**=**'python'**,** delimiter**=**'::'**)**

ratings\_mat **=** np**.**ndarray**(**

shape**=(**np**.max(**data**.**movie\_id**.**values**),** np**.max(**data**.**user\_id**.**values**)),**

dtype**=**np**.**uint8**)**

ratings\_mat**[**data**.**movie\_id**.**values**-**1**,** data**.**user\_id**.**values**-**1**]** **=** data**.**rating**.**values

normalised\_mat **=** ratings\_mat **-** np**.**asarray**([(**np**.**mean**(**ratings\_mat**,** 1**))]).**T

A **=** normalised\_mat**.**T **/** np**.**sqrt**(**ratings\_mat**.**shape**[**0**]** **-** 1**)**

U**,** S**,** V **=** np**.**linalg**.**svd**(**A**)**

movies\_index **=** movie\_data**[[**"title"**]].**set\_index**(**movie\_data**.**movie\_id**).**reindex**(range(**3953**)).**iloc**[**1**:].**title

users\_index **=** **list(range(**1**,** 6041**))**

**def** top\_cosine\_similarity**(**data**,** i**,** indexes**,** top\_n**=**10**):**

index **=** i **-** 1 # Movie/user id starts from 1

movie\_row **=** data**[**index**,** **:]**

magnitude **=** np**.**sqrt**(**np**.**einsum**(**'ij, ij -> i'**,** data**,** data**))**

similarity **=** np**.**dot**(**movie\_row**,** data**.**T**)** **/** **(**magnitude**[**index**]** **\*** magnitude**)**

sdata **=** pd**.**Series**(**similarity**,** index**=**indexes**).**sort\_values**(**ascending**=False)**

**return** sdata**.**iloc**[:**top\_n**]**

**def** get\_movie\_recommendation**(**movie\_id**,** k**=**50**,** top\_n**=**10**):**

sliced **=** V**.**T**[:,** **:**k**]** # representative data

results **=** top\_cosine\_similarity**(**sliced**,** movie\_id**,** movies\_index**,** top\_n**)**

name **=** movie\_data**[**movie\_data**.**movie\_id **==** movie\_id**].**title**.**values**[**0**]**

**print(**f"Recommendations for movie {name}"**)**

**print(**results**)**

**def** get\_user\_recommendation**(**user\_id**,** k**=**50**,** top\_n**=**10**):**

sliced **=** U**[:,** **:**k**]** # representative data

results **=** top\_cosine\_similarity**(**sliced**,** user\_id**,** users\_index**,** top\_n**)**

**print(**f"Recommendations for user {user\_id}"**)**

**print(**results**)**

get\_user\_recommendation**(**1**)**

**RESULTS OF THE PROGRAM WORK**

The input array is:

1::1193::5

1::661::3

1::914::3

1::3408::4

1::2355::5

1::1197::3.

The input user\_id is: 1

Output array:

Recommendations for user 1

1 1.000000

2017 0.952842

3969 0.905480

5985 0.898790

1953 0.891082

65 0.855654

2081 0.829714

4097 0.803258

4545 0.786801

2593 0.784218

dtype: float64

**CONCLUSIONS**

I got acquainted with the topic of laboratory work.

Have acquired relevant work skills.

An appropriate test program has been developed.

I learned how to handle big data, process and analyze it.