

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**  
Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
«\_\_\_» \_\_\_\_\_ 2022 р.

**БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА**  
**РОБОТА**

**РЕКОМЕНДАЦІЙНА СИСТЕМА З ПІДБОРУ ФІЛЬМІВ**  
**ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО**  
**НАВЧАННЯ**

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 401.21810111**

*Виконала студентка 4-го курсу, групи 401*  
\_\_\_\_\_ *М. Р. Бузінська*  
«\_\_\_» \_\_\_\_\_ 2022 р.

*Керівник: канд.техн.наук, доцент,*  
\_\_\_\_\_ *Г. В. Кондратенко*  
«\_\_\_» \_\_\_\_\_ 2022 р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Рівень вищої освіти **бакалавр**  
Спеціальність **122 «Комп'ютерні науки»**  
*(шифр і назва)*  
Галузь знань **12 «Інформаційні технології»**  
*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**З А В Д А Н Н Я**

**на виконання кваліфікаційної роботи**

Видано студенту групи 401 факультету комп'ютерних наук Бузінської Марії Романівни.

1. Тема кваліфікаційної роботи «Рекомендаційна система з підбору фільмів за допомогою методів машинного навчання».

Керівник роботи: канд.техн.наук, доцент, Г. В. Кондратенко

Затв. наказом Ректора ЧНУ ім. Петра Могили від « \_\_\_\_ » \_\_\_\_\_ 2022 \_\_\_\_ р. № \_\_\_\_\_

2. Строк представлення кваліфікаційної роботи студентом «\_30\_»  
\_травня\_ 2022\_\_ р.

3. Вхідні (початкові) дані до роботи: дані згідно з предметною областю та постановленою задачею.

Очікуваний результат: рекомендаційна система з підбору фільмів за допомогою методів машинного навчання.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– дослідження типів рекомендаційних систем та існуючі рішення;

- збір та обробка даних для тренування моделі;
- проведення експериментів тренування моделі;
- оцінка якості результатів;
- створення системи взаємодії з рекомендаційною системою

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Аналіз забезпечення вимог охорони праці у приміщенні серверної кімнати»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис
Спеціальна частина з охорони праці	ст. викладач Макарова О.В.	

Керівник роботи канд.техн.наук, доцент, Г. В. Кондратенко  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання Бузінська М. Р.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Дата видачі завдання « 23 » \_\_\_\_\_ листопада \_\_\_\_\_ 2021 р.

## КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: «Рекомендаційна система з підбору фільмів за допомогою методів машинного навчання».

№	Найменування роботи	Початок	Закінчення	Примітки
1	Затвердження завдання на кваліфікаційну роботу	23.11.2021	23.11.2021	Виконано
2	Дослідження літератури та матеріалів на задану тему	24.11.2021	05.01.2022	Виконано
3	Аналіз технічного завдання	05.01.2022	10.02.2022	Виконано
4	Проектування структури та алгоритму роботи розроблювальної системи	11.02.2022	03.03.2022	Виконано
5	Розробка компонентів системи	04.03.2022	28.03.2022	Виконано
6	Розгортання та тестування розробленої системи	29.03.2022	16.04.2022	Виконано
7	Виконання графічної частини	17.04.2022	21.04.2022	Виконано
8	Написання пояснювальної записки	22.04.2022	29.05.2022	Виконано
9	Представлення закінченої роботи на перевірку	30.05.2022	30.05.2022	Виконано
10	Передзахист кваліфікаційної роботи	31.05.2022	31.05.2022	Виконано
11	Захист кваліфікаційної роботи	28.06.2022	28.06.2022	Виконано

Розробив студент Бузінська М.Р. \_\_\_\_\_  
(прізвище та ініціали) (підпис)

Керівник роботи канд. техн. наук, доцент Г. В. Кондратенко \_\_\_\_\_  
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

«\_\_\_» \_\_\_\_\_ 2022 р.

# АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студентки групи 401

ЧНУ ім. Петра Могили

Бузінської Марії Романівни

«Рекомендаційна система з підбору фільмів за допомогою методів  
машинного навчання»

Дана кваліфікаційна робота присвячена розробці рекомендаційної системи з підбору фільмів за допомогою методів машинного навчання на основі оцінок та вподобань. Основним завданням в роботі є аналіз готових рішень рекомендаційних систем, та створення системи для побудови прогнозів рекомендацій фільмів для користувача.

**Метою** роботи є спрощення процесу підбору фільмів за допомогою методів машинного навчання.

**Об'єкт дослідження** – процеси створення рекомендацій на основі оцінок та вподобань.

**Предмет дослідження** – методи та технології створення рекомендаційної системи за допомогою машинного навчання.

Робота складається з фахового розділу і спеціальної частини з охорони праці. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі розглядається аналіз предметної області та алгоритмів машинного навчання для розв'язку поставленої задачі.

У другому розділі описано методи, алгоритми та інформаційні технології для вирішення поставленої задачі.

У третьому розділі описано засоби розробки для реалізації рекомендаційної системи.

Ключові слова: рекомендаційні системи, прогнозування, колаборативна фільтрація, контентна фільтрація, Machine Learning.

Пояснювальна записка складається з 4 розділів загальним обсягом 79 сторінок, містить 43 використаних джерела.

## **ABSTRACT**

of the bachelor's qualification work of the student of group 401 of Petro Mohyla  
Black Sea National University

Buzinska Mariia Romanivna

Topic «Recommended system for the selection of films using machine  
learning methods»

This thesis is devoted to the development of a recommendation system for the selection of films using machine learning methods based on grades and preferences. The aim of the work is to analyze ready-made solutions of recommendation systems, and to create an application for building predictions of film recommendations for the user.

**The aim** of the work is to simplify the process of selecting films with the help of machine learning methods.

**The object** of research is the process of making recommendations based on assessments and preferences.

**The subject** of research is methods and technologies of creating a recommendation system with the help of machine learning.

The work consists of a professional section and a special section on labor protection. The explanatory note consists of an introduction, four chapters, conclusions and appendices.

The first section discusses the analysis of the subject area and machine learning algorithms to solve the problem.

The second section describes the methods, algorithms and information technologies for solving the problem.

The third section describes the development tools for the implementation of the recommendation system.

**Keywords:** recommendation systems, forecasting, collaborative filtering, content filtering, Machine Learning.

## ЗМІСТ

ВСТУП.....	6
1 ПОСТАНОВКА ЗАДАЧІ ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ .....	8
1.1 Актуальність та проблеми підбору фільмів.....	8
1.2 Вимоги до рекомендаційної системи .....	9
1.3 Аналіз існуючих підходів щодо надання рекомендацій .....	11
1.4 Особливості предметної області.....	18
1.5 Аналіз аналогічних рекомендаційних систем.....	21
Висновки до розділу 1.....	22
2 МОДЕЛІ, МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ .....	23
2.1 Використання Java в розробці системи .....	23
2.2 Методи створення рекомендаційної системи.....	24
2.3 Java та штучний інтелект.....	27
2.4 Java та машинне навчання.....	29
3 ЗАСОБИ РОЗРОБКИ ДЛЯ РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ..	33
3.1 Архітектура рекомендаційної системи .....	33
3.2 Середовище розробки IntelliJ IDEA .....	35
3.3 Взаємодія користувача з рекомендаційною системою .....	36
3.4 Засоби розробки інтерфейсу користувача .....	38
3.5 Оцінки точності для рекомендаційних алгоритмів .....	39
3.6 Обґрунтування вибору програмної реалізації.....	39
3.7 Вимоги до рекомендаційної системи .....	40
3.8 Вимоги до програмного забезпечення .....	40



3.9	Проектування дизайну інтерфейсу користувача.....	40
3.10	Системні вимоги та інсталяція.....	41
3.11	Клієнтський інтерфейс.....	42
	Висновки до розділу 3.....	44
4	СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ .....	46
4.1	Вступ.....	46
4.2	Опис обраного виробничого приміщення, робочих місць, їх обладнання та складання вихідних даних для кількісної оцінки умов праці.....	47
4.3	Інтегральна оцінка умов праці в обраному виробничому приміщенні .....	51
4.4	Оцінка ефективності заходів щодо покращення умов праці.....	55
	Висновки до розділу 4.....	61
	ВИСНОВКИ.....	62
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
	ДОДАТОК А.....	68

## **ПЕРЕЛІК СКОРОЧЕНЬ**

ДСТУ – Державний стандарт України

БКР – бакалаврська кваліфікаційна робота

ПЗ – програмне забезпечення

РС – рекомендаційна система

ШІ – штучний інтелект

CF – collaborative filtering

RMSE – Root Mean Square Error

MAE – Mean absolute error

ROC – Receiver Operating Characteristics

PRC – Precision Recall Curve

# Пояснювальна записка

до кваліфікаційної роботи

на тему:

## «РЕКОМЕНДАЦІЙНА СИСТЕМА З ПІДБОРУ ФІЛЬМІВ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ»

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 401.21810111**

**Виконав студент 4-го курсу, групи 401**

М.Р. Бузінська

(підпис, ініціали та прізвище)

«\_\_» \_\_\_\_\_ 2022 р.

**Керівник:** к.т.н., доцент, доцент кафедри ІС

(наук. ступінь, вчене звання)

Г.В. Кондратенко

(підпис, ініціали та прізвище)

«\_\_» \_\_\_\_\_ 2022 р.

## ВСТУП

В Інтернеті, де кількість варіантів величезна, потрібно фільтрувати, розставляти пріоритети та ефективно доставляти відповідну інформацію, щоб полегшити проблему інформаційного перевантаження, що створило потенційну проблему для багатьох користувачів Інтернету. Рекомендаційні системи вирішили цю проблему шляхом пошуку у великому обсязі динамічно генерованої інформації, щоб надати користувачам персоналізований вміст та послуги.

Рекомендаційні системи - це системи фільтрації інформації які вирішують проблему перевантаження інформації, фільтруючи життєво важливий фрагмент інформації з великої кількості динамічно генерованої інформації відповідно до уподобань користувача, інтересів або спостережуваної поведінки щодо фільмів [1]. Система рекомендацій має можливість передбачити вибір користувача, чи буде конкретний користувач віддавати перевагу фільму чи ні на основі профілю користувача.

Рекомендаційні системи корисні як для постачальників послуг, так і для користувачів. Вони зменшують витрати на пошук і відбір фільмів та серіалів на Інтернет-сайтах. Системи рекомендацій також довели, що вони вдосконалюють процес прийняття рішень та якість. В наукових бібліотеках, рекомендаційні системи підтримують користувачів, дозволяючи їм перейти за межі пошуку за каталогами.

Система рекомендацій визначається як стратегія прийняття рішень для користувачів у складних інформаційних середовищах. Крім того, система рекомендацій була визначена з точки зору електронної комерції як інструмент, який допомагає користувачам здійснювати пошук у записах знань, пов'язаних із інтересами та уподобаннями користувачів.

Система рекомендацій була визначена як засіб сприяння та посилення соціального процесу з використанням рекомендацій інших для прийняття рішень, коли немає достатніх особистих знань чи досвіду альтернатив. Системи, що рекомендують, вирішують проблему перевантаження інформації, з якою зазвичай

стикаються користувачі, надаючи їм персоналізований ексклюзивний вміст та рекомендації щодо вибіру. Нещодавно були розроблені різні підходи до побудови систем рекомендацій, які можуть використовувати спільну фільтрацію [2], фільтрацію на основі вмісту або гібридну фільтрацію.

Техніка колаборативної фільтрації [4] є найбільш зрілою та найбільш часто застосовуваною.

Саме рекомендаційна система з підбору фільмів сприятиме економії часу при підбиранні фільмів за певними критеріями.

**Мета бакалаврської кваліфікаційної роботи** – спрощення процесу підбору фільмів за допомогою методів машинного навчання.

**Об'єкт дослідження** – процеси створення рекомендацій на основі оцінок та вподобань.

**Предмет дослідження** – методи та технології створення рекомендаційної системи за допомогою машинного навчання.

**Завдання:**

1. Постановка задачі побудови рекомендаційної системи та дослідження її типів;
2. Окреслення існуючих технологій для вирішення поставленої задачі;
3. Дослідження засобів розробки рекомендаційної системи;
4. Оцінка якості результатів;
5. Створення системи взаємодії з рекомендаційною системою.

# 1 ПОСТАНОВКА ЗАДАЧІ ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

## 1.1 Актуальність та проблеми підбору фільмів

Чи знайома вам ситуація: вирішили провести вечір удома і подивитися якесь кіно в хорошій компанії, але, спробувавши визначитися, яке – провели за вибором стільки часу, що на кіно його не залишилося – чи зникло бажання – чи все-таки почали щось дивитись, але настрої був уже не той?

Більшість людей списують цю проблему на свою недостатню поінформованість про світ кіно, і намагаються її вирішити за допомогою різних добірок та рейтингів, або запитуючи поради – а бізнеси, у свою чергу, намагаються зробити те саме, пропонуючи користувачам добірки та рейтинги [5].

Рекомендаційні системи займають важливе місце для таких відомих компаній як Amazon.com, YouTube, Netflix, Spotify, LinkedIn, Facebook, TripAdvisor, Last.fm, IMDb (Інтернет-база фільмів), Google та Yandex. Більше того, багато медіа компаній зараз розробляють рекомендаційні системи як частину свого сервісу, яку вони надають своїм користувачам. Компанія Netflix, онлайн провайдер потокового відео, нагородила призом в мільйон доларів команду, яка перша змогла значно покращити продуктивність їхньої рекомендаційної моделі.

Відповідно до досліджень проведених компанією McKinsey, більше 75% контенту, який дивились користувачі компанії Netflix, був саме запропонований їхньою рекомендаційною системою. Система допомагає споживачам знайти цікавий контент, який вони хочуть подивитись, а також допомагає компанії економити на витратах маркетингу.

Торгівельний гігант Amazon заявив, що 35% свого доходу він отримав завдяки їхнім рекомендаційним системам. Перш за все, вони показують товари, які часто купують з тими товарами, які у них вже є в кошику. Також вони показують товари, які схожі до тих, які вони щойно переглянули, та товари які є новими версіями до тих, які вони вже купили. Цікаво ще те, що вони вони використовують

рекомендаційні системи не лише на сайті, але і в email розсилках, які показують значний приріст у продажах [3].

Одним із основних методів вирішення проблеми інформаційного перевантаження є рекомендаційні системи, які забезпечують автоматизоване та персоналізоване пропонування інформації для споживачів.

Дана дипломна робота присвячена дослідженню методів побудови рекомендаційних систем. Метою роботи є створення рекомендаційної системи фільмів на основі оцінок та вподобань. Для створення актуальної системи був проведений широкий аналіз існуючих підходів до створення систем, що прогнозують рекомендації, та реалізація додатку із вбудованою системою рекомендацій.

Завданням даної роботи є реалізація колаборативного методу для побудови рекомендаційної системи, яка зможе рекомендувати фільми користувачам. Для досягнення поставленої мети в роботі були виділені наступні завдання:

- постановка задачі побудови рекомендаційної системи та дослідження її типів;
- окреслення існуючих технологій для вирішення поставленої задачі;
- дослідження засобів розробки рекомендаційної системи;
- оцінка якості результатів;
- створення системи взаємодії з рекомендаційною системою.

Розроблений продукт можна використовувати як швидку та єдину платформу для прогнозу рекомендацій фільмів. За рахунок побудови рекомендаційної системи з користувацьким інтерфейсом реалізується універсальність та зручність використання.

## **1.2 Вимоги до рекомендаційної системи**

Рекомендаційна система – підклас системи фільтрації інформації, яка буде рейтинговий перелік об'єктів (фільми, музика, книги, новини, вебсайти), яким користувач може надати перевагу. Для цього використовується інформація, яку

надав користувач. Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і колаборативна фільтрація [4].

Рекомендаційна система повинна містити інформацію, яка буде зручною та корисною для кожного, хто користуватиметься нею.

Всю інформацію користувач повинен знаходити максимально швидко, а не блукати по системі в пошуках потрібного.

У даному випадку рекомендаційна система використовує принцип, що якщо користувачам подобалися одні й самі, чи дуже схожі, об'єкти, то й далі їх поведінка збігатися. Відповідно, при даному підході можна не використовувати великого обсягу інформації про об'єкти, а виходити тільки з оцінок користувача об'єктів. Далі наведемо наочний приклад. Нехай є користувачі  $U_1$  та  $U_2$ , і користувач  $U_1$  подивився фільми  $M_1, M_2, M_3, M_4$ , а користувач  $U_2$  подивився фільми  $M_2, M_3, M_4, M_5$ . Оскільки для обох користувачів 3 із 4 фільмів загальні ( $M_2, M_3, M_4$ ), то користувачеві  $U_1$  можна порекомендувати фільм  $M_5$ , а користувачеві  $U_2$  – фільм  $M_1$ .

Розглянемо відмінний від описаних вище підхід. Великий наголос буде робитися на способи обробки інформації та засіб отримання передбачень оцінок користувача. Однак, у згаданих способах фільтрації закладена головна ідея, якої варто дотримуватися при створенні рекомендаційних систем, а саме пошук користувачів зі схожими перевагами та рекомендація їм фільмів, які подобаються одному і не переглянуті іншим, а також визначення схожості фільмів для рекомендації людям схожого на переглянуте ними кіно.

Говорячи про практичне застосування, існує безліч інтернет ресурсів з інформацією про фільми, на яких користувачі можуть виставляти оцінки переглянутим фільму. Щоб не витратити час на пошуки наступного фільму для перегляду, користувачі таких сайтів потребують рекомендаційної системи. Відповідно, створена в рамках даної роботи рекомендаційна система може використовуватись на згаданих сайтах як основне ядро для функціональності їх системи показу рекомендацій.



Також, торкаючись теми застосування отриманої в цій роботі системи, варто зазначити, що, враховуючи різні вже наявні бази даних фільмів та користувачів на інтернет-ресурсах про кіно, де якраз і можливе використання рекомендаційної системи, будь-яка така система потребуватиме, як мінімум, у донавченні або перенастроюванні для коректної роботи, тобто для хороших рекомендацій користувачам з конкретною базою даних конкретного сайту. І, відповідно, рекомендаційна система досить легко може масштабуватися за допомогою оновлення баз даних новими користувачами та фільмами. Залежно від системи може знадобитися деяка обробка нових доданих до бази даних.

### **1.3 Аналіз існуючих підходів щодо надання рекомендацій**

Рекомендації бувають двох типів: персоналізовані та не-персоналізовані. Прикладом не персоналізованих - це рекомендувати найпопулярніші фільми, або рекомендувати відштовхуючись від бізнес-завдань. Серед підходів персоналізованої рекомендації розрізняють content-based (базовані на контенті), collaborative filtering (колаборативну фільтрацію) та гібридну техніку, яка поєднує два попередні методи. Загальний принцип content-based методів - знайти спільні характеристики між фільмами, які отримали схвальну оцінку від користувача, а потім рекомендувати для цього користувача нові фільми, які також мають ці характеристики. Рекомендаційні системи, які базуються лише на контенті зазвичай мають проблеми аналізу обмеженого вмісту та надмірної спеціалізації [6].

Аналіз обмеженого вмісту настає коли система має обмежену кількість інформації про користувачів та фільми. Наприклад, вимога про конфіденційність приватних даних не дозволяє їх використовувати для аналізу, або докладна інформація про фільм є недоступною, дорогою для зібрання, або складною для аналізу - наприклад картинки, музика, тощо. Іншою проблемою є те, що контент фільмів зазвичай є недостатнім, щоб визначити його цікавість. Надмірна спеціалізація, з іншого боку, є побічним ефектом того, як рекомендують content-

based методи нові фільми, коли прогнозована оцінка є високою, якщо фільм є схожим до того, який сподобався користувачу. Наприклад, в додатку для рекомендацій фільмів, система може рекомендувати користувачу фільми такого ж жанру, або ті які мають однакових акторів з тими фільмами, які користувач вже подивився. Однак система не спроможна порекомендувати фільми, які відрізняються від попередніх і які також були б цікаві користувачу. Замість того щоб залежати від контентної інформації, методи колаборативної фільтрації використовують інформацію про оцінки від інших користувачів про товари в системі. Ключовою ідеєю є те, що оцінка певного юзера для нового фільму буде схожою з тою, яку поставив інший користувач, якщо вони обидва оцінили інші фільми подібними оцінками. Колаборативна фільтрація змогла подолати певні обмеження, які має content-based. Наприклад, фільми, контент яких є відсутнім або важко доступним, можуть бути рекомендовані базуючись на відгуках інших користувачів.

Більше того, колаборативні рекомендації базуються на якості фільмів які оцінені іншими, ніж покладатись на контент який може бути поганим індикатором цікавості фільму. Також колаборативна фільтрація може рекомендувати різноманітні фільми з різним контентом, оскільки інші користувачі проявили інтерес до цих різних фільмів. Підходи колаборативної фільтрації можуть бути згруповані на два загальні класи базовані на знаходження сусідів та базовані на навчанні моделі. В підході основою якого є знаходження сусідів, оцінки користувача щодо фільмів, які збережені в системі, безпосередньо використовуються під час прогнозування оцінок для нових фільмів. Це може бути здійснено двома способами відомими як user-based та item-based рекомендаціями. User-based системи оцінюють інтерес певного користувача для фільму використовуючи оцінки для цих фільмів від інших користувачів, які називають сусідами та мають схожі моделі поведінки оцінювання. Сусіди певного користувача є зазвичай користувачі, оцінки яких найбільше корелюють з оцінками даного користувача. Item-based підхід, з іншого боку, прогнозує оцінки

користувача для фільму базуючись на оцінках користувача для схожих фільмів. В таких підходах, два фільми є схожими якщо декілька користувачів системи оцінили ці фільми подібними оцінками.

В протипагу методам на знаходження сусідів, які використовують збережені оцінки безпосередньо під час прогнозування, методи базовані на моделях використовують оцінки щоб навчити рекомендувати модель. Важливі характеристики користувачів та фільмів зберігаються в множині параметрів моделі, які отримані під час навчання моделі на тренувальних даних і можуть бути використані для прогнозування нових оцінок.

Різні типи рекомендаційних систем використовують різні підходи до цих питань. Загалом, існує чотири типи рекомендаційних систем:

- Системи рекомендацій на основі вмісту – цей тип рекомендаційної системи орієнтований, ну, на зміст. Це означає, що вони використовують лише функції та інформацію з елементів і на їх основі створюють рекомендації для користувача. Вони не враховують інформацію від інших користувачів.

- Системи рекомендацій для спільної фільтрації. Найбільша сила рекомендаційних систем полягає в тому, що вони можуть пропонувати елементи для користувачів на основі їхньої поведінки на певній платформі або на основі поведінки інших користувачів тієї ж платформи. Наприклад, Netflix пропонує вашій наступній серії гуляти, виходячи з серіалу, який ви дивилися раніше, але на основі серіалів, які користувачі дивилися та вподобали той самий вміст, що й ви.

- Системи рекомендацій, засновані на знаннях. Цей тип рекомендаційної системи використовує чіткі знання про переваги користувача, елементи та/або критерії рекомендацій. У цьому сценарії системи рекомендацій запитують користувача про їхні переваги та на основі цього відгуку створюють рекомендації.

- Системи рекомендацій щодо гібридних рішень – ми часто використовуємо комбінацію всіх типів для деяких індивідуальних рішень.

Існують різноманітні підходи до створення моделей для задачі рекомендацій. Деякі з них розглянемо детальніше пізніше. На завершення, щоб подолати певні обмеження методів базованих на контенті та колаборативної фільтрації, використовують гібридні рекомендаційні методи, які поєднують кращі характеристики обох підходів. Поєднати ці методи можна різноманітними способами, наприклад, поєднавши окремі списки рекомендацій в один, або додавання контентної інформації в моделі колаборативної фільтрації. Декілька досліджень показали, що використання гібридних моделей забезпечило більш точні рекомендації ніж чисті content-based та колаборативні методи, особливо коли наявні лише декілька оцінок. Оскільки електронна комерція (e-commerce) набирає популярності, важливим завданням є допомогти користувачам легко відсортувати серед великої різноманітності запропонованих фільмів ті, які найбільше відповідають їхнім бажанням.

Одним з засобів вирішення поставленого завдання є рекомендаційні системи, які зараз активно розробляються та вдосконалюються. Ці системи забезпечують користувачів персоналізованими рекомендаціями фільмів або сервісами, які бажано б підходили їхнім інтересам та потребам.

Рекомендаційні системи базуються на двох різних стратегіях (або поєднанню них). Підхід базований на контенті (Content based) створює профіль для кожного користувача та фільму з їхніх характеристик. Наприклад, профіль фільму може містити ознаки жанру, акторів, касової популярності, тощо. Профіль користувача може містити демографічну інформацію або відповіді на запропоновані запитання. Отримані профайли надають змогу програмам асоціювати користувачів з відповідними фільмами. Однак, цей підхід потребує збирання інформації, яка може не бути в наявності або складною для накопичення. Альтернативна стратегія потребує лише історії поведінки користувача без створення явного профілю. Такий підхід відомий як Collaborative Filtering (CF) суспільна фільтрація. CF аналізує взаємозв'язки між користувачами та залежності серед фільмів, щоб знайти нові користувач-фільм асоціації. Наприклад, деякі CF системи знаходять групи фільмів

завдяки тому, що вони були оцінені однаково або вподобані юзерами з схожою історією оцінок.

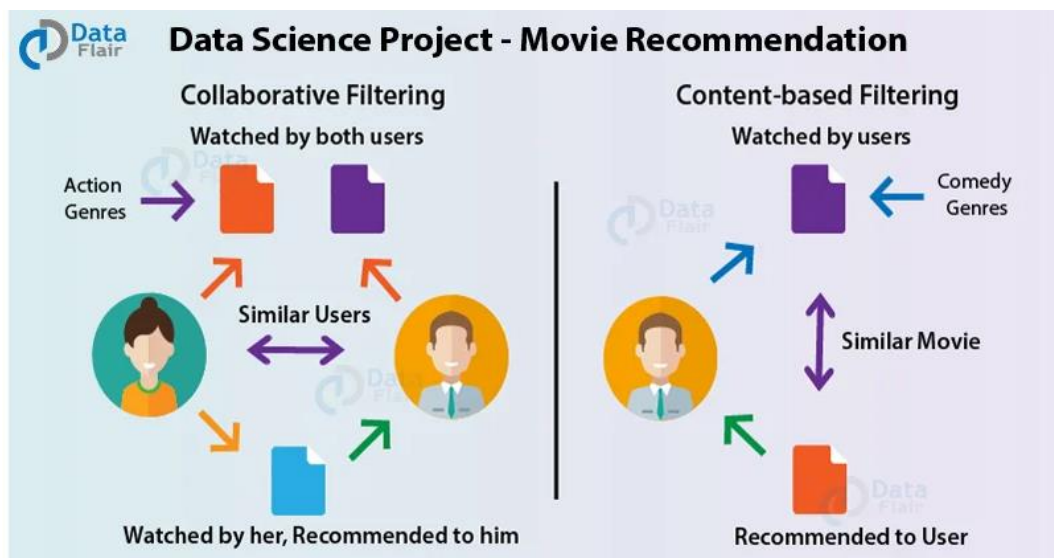


Рисунок 1.1 – Collaborative Filtering and Content - based Filtering [41]

Що таке рекомендаційна система?

Система рекомендацій надає користувачам пропозиції через процес фільтрації, який базується на перевагах користувача та історії перегляду. В якості вхідних даних береться інформація про користувача. Інформація береться з вхідних даних у вигляді даних перегляду. Рекомендаційна система – це платформа, яка надає своїм користувачам різноманітний контент на основі їхніх уподобань та вподобань. Система рекомендацій приймає інформацію про користувача як вхідну інформацію. Рекомендаційна система – це реалізація алгоритмів машинного навчання.

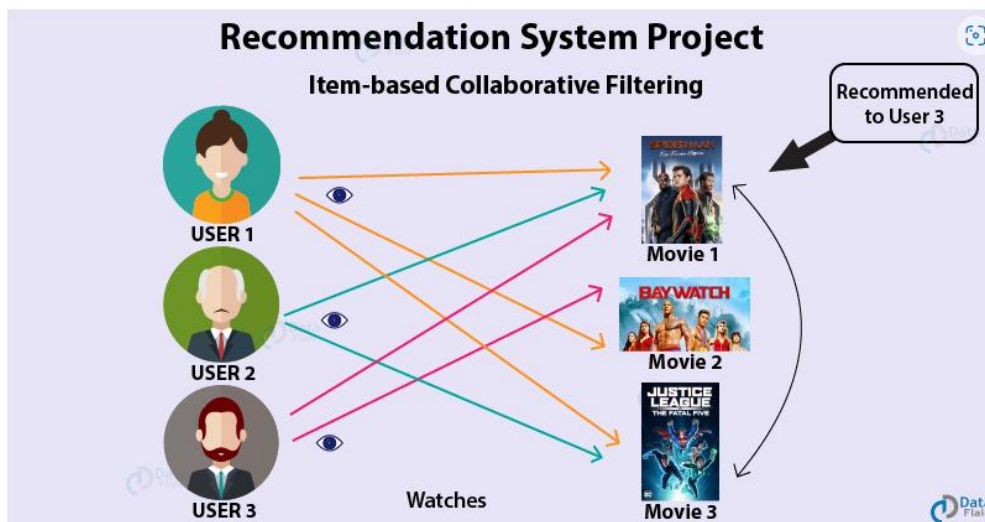


Рисунок 1.2 – Наглядний приклад рекомендації на основі спільної фільтрації [42]

Система рекомендацій також знаходить схожість між різними продуктами. Наприклад, система рекомендацій Netflix надає вам рекомендації щодо фільмів, подібних до тих, які дивилися раніше. Крім того, існує спільна фільтрація вмісту, яка надає вам рекомендації щодо інших користувачів, які можуть мати подібну історію перегляду або вподобання. Існує два типи рекомендаційних систем – рекомендаційна система на основі вмісту та рекомендація спільної фільтрації. Спільна фільтрація на основі елементів – це алгоритм на основі моделі для створення рекомендацій. В алгоритмі подібність між різними елементами в наборі даних обчислюється за допомогою одного з ряду показників подібності, а потім ці значення подібності використовуються для прогнозування оцінок для пар користувач-елемент, яких немає в наборі даних.

### Типи колаборативної фільтрації:

#### *Заснований на пам'яті*

Цей підхід використовує дані про рейтинг користувача для розрахунку схожості між користувачами або предметами. Він використовується для вироблення рекомендацій. Це був початковий підхід, що використовувався в багатьох торгових системах. Він ефективний і простий у реалізації. Типовими прикладами такого підходу є CF і засновані на виробі/користувачеві топ-N рекомендації. Наприклад, у підходах, заснованих на користувачеві, вартість оцінки,

яку користувач  $u$  дає виробу « $i$ » розрахована як сукупність схожих оцінок виробу іншими користувачами.

#### *Заснований на сусідстві*

Алгоритм, заснований на сусідстві, обчислює подібність двох користувачів або виробів, виробляє прогноз для користувача, приймаючи середнє зважене всіх оцінок. Обчислення схожості між виробами або користувачами є важливою частиною цього підходу. Багаторазові заходи, такі як кореляції Пірсона і схожість, заснована на скалярному добутку, використовується для цього.

#### *Заснований на моделі*

Даний підхід надає рекомендації, вимірюючи параметри статистичних моделей для оцінок користувачів, побудованих за допомогою таких методів як, метод баєсовських мереж, кластеризації, латентно-семантичної моделі, такі як сингулярний розклад, імовірнісний латентно-семантичний аналіз, прихований розподіл Дирихле і марковський процес вирішування на основі моделей. Моделі розробляються з використанням інтелектуального аналізу даних, алгоритмів машинного навчання, щоб знайти закономірності на основі навчальних даних. Число параметрів в моделі може бути зменшено в залежності від типу за допомогою методу головних компонент.

Цей підхід є більш комплексним і дає більш точні прогнози, оскільки допомагає розкрити латентні фактори, що пояснюють спостережувані оцінки.

Даний підхід має ряд переваг. Він обробляє розріджені матриці краще, ніж підхід заснований на сусідстві, що в свою чергу допомагає з масштабістю великих наборів даних.

Недоліки цього підходу полягають в «дорогому» створенні моделі [6]. Необхідний компроміс між точністю і розміром моделі, тому що можна втратити корисну інформацію у зв'язку із скороченням моделей.

#### *Гібридний підхід*

Даний підхід об'єднує в собі підхід заснований на сусідстві і заснований на моделі. Гібридний підхід є найпоширенішим при розробці рекомендаційних систем

для комерційних сайтів, так як він допомагає подолати обмеження початкового оригінального підходу (заснованого на сусідстві) і поліпшити якість прогнозів. Цей підхід також дозволяє подолати проблему розрідженості даних і втрати інформації. Однак даний підхід складний і дорогий у реалізації та застосуванні [43].



Рисунок 1.3 – Типи колаборативної фільтрації

#### 1.4 Особливості предметної області

Оцінки вподобань користувачем фільмів можуть бути числові значення відомі як рейтинги (наприклад зірочки 1-5), бінарні значення (сподобалось/не сподобалось). Більше того, відгуки користувача можуть бути отримані в явному (explicit) вигляді, наприклад введені користувачем в систему, або неявні (implicit), отримані з історії переглядів, тощо. Оцінки отримані у явному вигляді - вважаються більш цінними, оскільки користувач особисто вказав, наскільки йому подобається даний фільм. З іншого боку, користувачі часто не залишають відгуки - це призводить до втрати важливої інформації, яка могла б покращити рекомендації. Тому з неявних відгуків можна отримати значно більше інформації про вподобання того чи іншого користувача.

Першим чинником, який слід врахувати при розробці РС, є область застосування, оскільки вона має значний вплив на алгоритмічний підхід, який необхідно прийняти. Монтанер [13] надає таксономію РС та класифікує існуючі додатки РС до певних доменів додатків. Виходячи з цих конкретних доменів додатків, розглянемо більш загальні класи доменів для найпоширеніших додатків



рекомендаційних систем:

- Розваги – рекомендації для фільмів, музики, ігор, тощо.
- Контент – персоналізовані газети, новини, рекомендації документів, веб сторінок в пошукових запитах, рекомендації онлайн курсів, а також фільтрація по категоріям для електронної пошти.
- Електронна комерція – рекомендації товарів для покупок, таких як книжки, телефони, ноутбуки, аксесуари, тощо.
- Послуги – рекомендації туристичних послуг, рекомендації експертів для консультації, рекомендації будинків для оренди або послуги знайомств.
- Соціальні мережі – рекомендація людей у соціальних мережах та рекомендації контенту в соціальних медіа, такі як твіти, канали Facebook, оновлення LinkedIn та інші.

### **Основні функції рекомендаційної системи.**

Сприяти економії часу при підбиранні фільмів за певними критеріями. Ця мета досяжна, оскільки рекомендовані фільми зазвичай задовольняють потребам та інтересам користувачів. Можливо користувач знайде цікавий йому фільм після запропонованих декількох рекомендацій.

Некомерційні організації мають подібні цілі, навіть тоді, коли для користувача це нічого не коштує. Наприклад, новинний портал хоче збільшити кількість переглянутих новин на сайті. Загалом, головною метою впровадження рекомендаційної системи є збільшення відсотку конверсії. Це можна побачити, порівнявши кількість користувачів які прийняли рекомендацію та спожили товар, в порівнянні до кількості простих відвідувачів, які просто шукали товари серед всієї інформації. Продавати більш різноманітні товари.

Іншою основною функцією рекомендаційних систем є надання користувачу можливості вибрати товари, які важко було б знайти без рекомендації. Наприклад, туристичні компанії зацікавлені в тому, щоб рекомендаційна система рекомендувала відвідувачам всі можливі туристичні місця, а не лише

найпопулярніші. Це може бути досить складно без рекомендаційних систем, оскільки компанія не може ризикувати рекламувати ті місця, які можуть не сподобатися користувачу. Саме тому, рекомендаційні системи пропонують або рекламують не дуже популярні місця саме для тих користувачів, яким потенційно вони можуть сподобатись. Збільшити задоволення користувача. Якщо рекомендовані товари будуть корисними, релевантними, та розроблена система буде мати зручний та привабливий вигляд, то користувачі будуть отримувати задоволення від використання системи.

Поєднання ефективних, точних рекомендацій та приємного користувацького інтерфейсу значно підвищать особисте ставлення користувача до системи. Це, свою чергу, збільшить корисність системи та більш ймовірно, що рекомендації будуть прийняті та спожиті.

Збільшити лояльність користувачів. Користувачі будуть більш відданим до веб сервісу, якщо під час відвідування, система розпізнає їх як минулих користувачів та буде поводитись з ними як з цінними відвідувачами. Це є стандартна особливість рекомендаційних систем, оскільки вони здійснюють рекомендації, на основі попередніх взаємодій користувача, такі як переглянуті фільми та залишені до них оцінки. Відповідно, чим довше користувач взаємодіє з веб сайтом, тим більш модель стає адаптованою до потреб користувача, отримує краще розуміння його уподобань, рекомендації стають більш персоналізовані.

Краще розуміння потреб користувача. Іншою важливою функцією рекомендаційної системи, яка може бути використана в багатьох інших програмах - це опис уподобань користувача, який збирається від користувачів у явному вигляді або прогнозується системою. Постачальник послуг може потім вирішити перевикористати ці знання для багатьох інших цілей, таких як покращити управління додавання нових фільмів та жанрів. Наприклад, в контексті туристичного бізнесу, управління напрямками в організації може вирішити рекламувати специфічні регіони для певних груп користувачів базуючись на аналізі інформації зібраної рекомендаційною системою [14].

## 1.5 Аналіз аналогічних рекомендаційних систем

Рекомендаційні системи вже інтегровані в безліч веб додатків, які широко використовуються кожен день мільйонами користувачів. Розглянемо приклади найбільших ресурсів, що використовують рекомендаційні механізми. Внаслідок пошуку вже існуючих аналогічних програмних систем та рішень було детально проаналізовано наступні реалізовані програмні продукти:

- LinkedIn – бізнес-орієнтована соціальна мережа. Вбудований рекомендаційний механізм пропонує користувачеві рекомендації людей, яких він, можливо, знає, вакансій, які могли б його залучити, груп, в які він міг би захотіти вступити, компаній, якими він міг би зацікавитися. Спеціалізована система колаборативної фільтрації LinkedIn заснована на технології Apache Hadoop;

- Amazon - одна з найбільших майданчиків інтернет-торгівлі - використовує рекомендації на основі контенту. Коли відвідувач вибирає для покупки будь-якої товар, Amazon на основі цього вихідного товару рекомендує відвідувачеві інші товари, придбані іншими користувачами (за допомогою матриці покупки наступного товару на основі його схожості з попередньою покупкою). Компанія Amazon запатентувала цей підхід під назвою item-to-item collaborative filtering (колаборативна фільтрація від елемента до елемента);

- Apple Music створює музичну «станцію» рекомендованих пісень, спостерігаючи, які групи і окремі треки користувач прослуховує на регулярній основі. Apple Music відтворює доріжки, що не присутні в бібліотеці користувача, але часто відтворюються іншими користувачами з аналогічними інтересами. Оскільки цей підхід використовує поведінку користувачів, він є прикладом спільної фільтрації;

- Spotify використовує метадані пісень і виконавців близько 400 атрибутів, наданих проектом Music Genome Project, щоб згенерувати «станцію», яка відтворює музику з подібними властивостями. Крім того, для

уточнення результатів «станції» використовується зворотний зв'язок від користувача, яка знецінює певні атрибути, коли користувачеві не сподобалася певна пісня і збільшує внесок інших атрибутів, коли користувачеві подобається пісня. Даний сервіс використовує контенторієнтований підхід;

— Netflix представила персоналізовані рекомендації щодо фільмів. Netflix використовує CineMatch, власну систему рекомендацій, котра мала середньоквадратичну помилку (RMSE) 0,9525 і закликала людей перевершити цей показник на 10%. Алгоритм CineMatch мав тривалий термін і виявився досить успішним у прогнозуванні фільмів, які сподобаються передплатникам [3].

Компанія Google також випустила статтю під назвою “Deep Neural Networks for YouTube Recommendations” [26] про рекомендаційну систему для іншого сервісу, а саме YouTube. Особливість описаної системи полягає в тому, що використовується дві незалежні нейронні мережі. Перша нейронна мережа на основі історії користувача з мільйона не переглянутих ним відео відбирає сотні кандидатів для рекомендації, застосовуючи метод найближчого сусіда на результат її передбачень.

Наступним кроком використовується інша нейронна мережа, яка навчена ранжувати відібрані відео з використанням більшого обсягу інформації про них.

### **Висновки до розділу 1**

У цьому розділі було досліджено існуючі методики рекомендаційних систем. Також у цьому розділі був проведений аналіз існуючих рішень комерційних сайтів та стрімінгових сервісів, де використовуються рекомендаційні системи для покращення користування користувачам. Серед гігантів сервісів в яких використовуються рекомендаційні системи, був виділений Netflix, як сервіс, що надає персоналізовані рекомендації до фільмів. Було досліджено актуальність та проблеми підбору фільмів, розглянуті вимоги до рекомендаційної системи, проведено аналіз існуючих підходів щодо надання рекомендацій та аналіз аналогічних рекомендаційних систем.

## 2 МОДЕЛІ, МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

### 2.1 Використання Java в розробці системи

Java – об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розроблення об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Java вплинула на розвиток J++, що розроблялась компанією «Microsoft». Роботу над J++ було зупинено через судовий позов «Sun Microsystems», оскільки ця мова програмування була модифікацією Java. Пізніше в новій платформі «Microsoft» .NET випустили J#, щоб полегшити міграцію програмістів J++ або Java на нову платформу. З часом нова мова програмування C# стала основною мовою платформи, перейнявши багато чого з Java. [9]

На сьогоднішній день з використанням цієї мови реалізуються різні програми: від невеликих десктопних застосунків до великих веб-сервісів, які щодня обслуговують мільйони користувачів.

Точність прогнозів, зроблених рекомендаційною системою, залежить від оцінок, що користувач розставляє у програмі. На основі вводу користувача програма створює вектор оцінок фільмів, а використовуючи бібліотеку машинного навчання, значення вектора та набір даних з оцінками фільмів можна знайти вибірку фільмів які скоріше за все теж будуть до вподоби користувачу. Існує кілька функцій оцінки подібності, таких як косинусна подібність, коефіцієнт кореляції Пірсона, тощо.

У цьому проєкті використана механізм, що базується на функції косинусної подібності `CosineSimilarity()`, для знаходження підходящих фільмів на основі порівняння набору оцінок користувача з оцінками інших користувачів.

## 2.2 Методи створення рекомендаційної системи

Для створення рекомендаційної системи на основі методу колаборативної фільтрації була вибрана мова програмування Java, що стрімко розвивається та має багато можливостей. Java дозволяє як натренувати модель нейронної мережі так і створити API, що дозволить інтегрувати систему прогнозу рекомендацій у інтерфейс користувача. Треба сказати, що сьогодні втілювати в життя рішення на базі тих чи інших технологій зі сфери ШІ може практично будь-який підкований Java-програміст, який має базові знання традиційних алгоритмів і концепцій Java. Для цього існує ціла маса відкритих бібліотек Java, які допоможуть реалізувати безліч досить складних функцій.

Слід лише відзначити, що ШІ в цілому – це досить широка область, і сьогодні ми поговоримо про Java-бібліотеки в найбільш популярних нішах ШІ, таких як обробка природної мови, машинне навчання, нейронні мережі тощо. До основних фреймворків створення нейронних мереж на мові програмування Java, можна віднести наступні:

- `Neuroph` – це бібліотека з відкритим вихідним кодом для створення нейронних мереж та використання машинного навчання, написана на Java. Користувачі `Neuroph` можуть створювати нейромережі через наданий

графічний інтерфейс або за допомогою Java-коду. На сайті Neuroph є документація, в якій також докладно пояснюється, що таке нейронна мережа насправді та як вона працює [15].

– Deerlearning4j – бібліотека з відкритим вихідним кодом створена для реалізації нейронних мереж на мові Java. Вона може використовувати не тільки потужність процесору, але й відеокарти. Має дуже гнучкий інтерфейс та може бути використана для більшості задач штучного інтелекту. Включає можливості не тільки створення нейронних мереж, але й підготовку та нормалізацію даних, швидкі структури для роботи з великими масивами даних. Однією величезною перевагою є величезна кількість вбудованих алгоритмів, функцій активації та методів обробки даних [12].

Це розподілена бібліотека глибокого навчання з відкритим вихідним кодом, яка має перевагу нових середовищ розподілених обчислень, таких як і . Деякі з його особливостей:

- комерційний рівень та відкритий вихідний код;
- приносить II до бізнес-середовища;
- детальний документ API;
- приклади проектів кількома мовами;
- інтегрований з Hadoop та Apache Spark [40].

Спираючись на постановку завдання, опишемо підхід до її вирішення, який розглядатиметься у цій роботі.

Для визначення відповідних рекомендацій фільмів необхідно проаналізувати наявні оцінки користувачів та інформація про фільми. Далі, на основі цього будуть робитися пропозиції для кожного не переглянутого фільму про те, з якою ймовірністю конкретний користувач вибере цей фільм для перегляду. Це буде реалізовано за допомогою алгоритмів машинного навчання. Деякі спеціалісти зробили спроби у створенні рекомендаційних систем на подібній основі.

Як базу інформації для рекомендаційної системи було обрано найпопулярніший і найширший набір даних «MovieLens» [39].

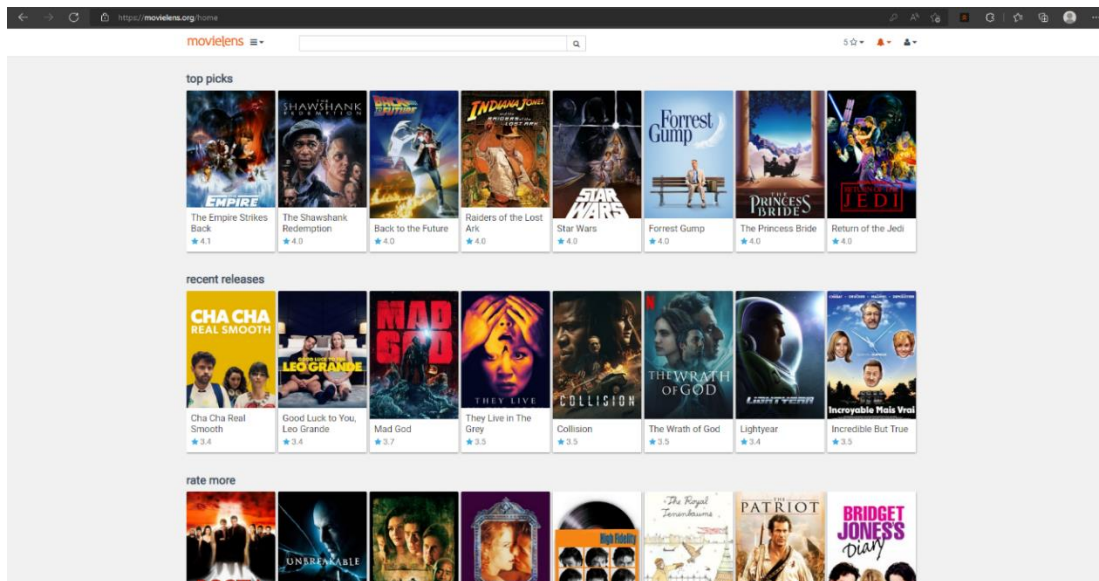


Рисунок 2.1 – Вигляд сайту «MovieLens»

Цей датасет має реальну інформацію і був спеціально зібраний дослідницькою лабораторією для рекомендаційних мереж, їх вивчення, аналізу, розробки та експериментів. «MovieLens» містить дані про 283228 користувачів та їх оцінки, яких у сумі 27753444. Наявні оцінки відносяться до 58098 фільмів, про які є короткі відомості: назви та список жанрів. Оцінки можуть приймати значення від 1 до 5 з кроком 0,5.

Основним файлом даного датасету є «ratings.csv». Він складається з 4 колонок: у перших двох колонках записані унікальні ідентифікатори користувача та фільму, у третій колонці міститься оцінка, яку цей користувач привласнив зазначеному фільму, а у четвертій колонці – ідентифікатор часу цієї події.



	userId	movieId	rating	timestamp
<b>1057</b>	10	3253	3.0	948881454
<b>1058</b>	10	3260	3.0	948882230
<b>1059</b>	10	3265	4.0	948882177
<b>1060</b>	11	48	3.0	1112135389
<b>1061</b>	11	158	3.5	1112135405
<b>1062</b>	11	527	4.0	1112135526
<b>1063</b>	11	1193	4.5	1112135548
<b>1064</b>	11	1282	4.5	1112135458
<b>1065</b>	11	1639	4.0	1112135379
<b>1066</b>	11	1722	3.5	1112135463

Рисунок 2.2 – Фрагмент вмісту файлу «ratings.csv»

Оскільки «MovieLens» містить малу кількість інформації про фільми, то вона була доповнена вручну. У цьому наборі міститься інформація про 3144 фільми, випущені до червня 2021 року включно. Датасет складається з файлів формату .csv, які включають всю доступну інформацію про фільми. Основні дані містяться у файлі «ratedmoviesfull.csv». Там записані назви, описи, слогани, жанри, країни виробництва та багато іншого для кожного з фільмів.

Для роботи з даними використовувався Java.ML для мови програмування Java.

### 2.3 Java та штучний інтелект

Java – об'єктно-орієнтована мова програмування. У сфері ШІ використовується досить багато мов програмування, але Java безперечно одна з найпопулярніших. Зокрема, Java застосовують при створенні рішень для машинного навчання, нейронних мереж, алгоритмів пошуку, генетичного

програмування та мультироботехнічних систем.

Такі властивості, як об'єктно-орієнтованість і масштабованість, обов'язкові для ІІІ-проектів, а тому Java підходить їм якнайкраще. Оскільки сьогодні ІІІ-технології вже активно застосовуються бізнесом, дуже потрібні можливості Java, що дозволяють створити єдину версію програми, яка працюватиме на кількох платформах [16].

Ще однією перевагою Java у програмуванні ІІІ є здатність кодувати різні типи алгоритмів. Та й інші сильні сторони та переваги Java, які ми вже не раз згадували, потрібні у сфері ІІІ-розробки. Серед них – простота створення та налагодження коду, спрощена робота з великомасштабними проектами, спрощена візуалізація, а також наявність Swing та SWT (Standard Widget Toolkit) [17]. Ці функції дозволяють створювати привабливу складну графіку та інтерфейси.

### ІІІ бібліотеки на Java

Треба сказати, що сьогодні втілювати в життя рішення на базі тих чи інших технологій зі сфери ІІІ може практично будь-який підкований Java-програміст, який має базові знання традиційних алгоритмів і концепцій Java. Для цього існує ціла маса відкритих бібліотек Java, які допоможуть реалізувати безліч досить складних функцій.

Слід лише відзначити, що ІІІ в цілому – це досить широка область, і сьогодні ми поговоримо про Java-бібліотеки в найбільш популярних нішах ІІІ, таких як обробка природної мови, машинне навчання, нейронні мережі тощо.

### **Експертні системи**

В ІІІ експертними системами називаються комп'ютерні системи, які емулюють здатність людини приймати рішення.

#### *Apache Jena*

Apache Jena – це відкритий Java-фреймворк для створення семантичних мереж та додатків пов'язаних даних на основі моделі RDF. На офіційному сайті Apache Jena можна знайти докладний посібник із використання цього фреймворку з коротким описом специфікації RDF [18].

### *PowerLoom*

PowerLoom (PowerLoom Knowledge Representation and Reasoning System) – це платформа для створення інтелектуальних, що базуються на знаннях додатків. PowerLoom використовує мову уявлення знань (варіант KIF), основу якого лежить природний механізм дедукції. Цей механізм вміє створювати логічний ланцюжок для отримання висновків на основі фактів та правил, зазначених у базі знань [19].

### *D3web*

D3web – це семантичний механізм міркувань з відкритим вихідним кодом для розробки, тестування та застосування фактичних знань у конкретній ситуації. D3web включає цілу низку комплексних алгоритмів. На офіційному сайті можна знайти короткий вступ до платформи з безліччю прикладів її застосування та документацією [20].

## **2.4 Java та машинне навчання**

Машинне навчання еволюціонувало в галузі штучного інтелекту, мета якої – створення машин, здатних наслідувати людський інтелект.

Машинне навчання – це потужний метод, що дозволяє вчитися на прикладах та досвіді. Це дозволяє програмним додаткам вчитися на даних і ставати точнішими при прогнозуванні результатів без втручання людини або без програмування. Тому замість того, щоб писати весь код вам просто потрібно передати дані, і алгоритм побудує логіку на основі ваших даних.

Алгоритм машинного навчання – це еволюція звичайного алгоритму. Це робить ваші програми «розумнішими», дозволяючи їм автоматично вчитися на наданих даних. Алгоритм в основному ділиться на два етапи: Підготовка та Тестування.

Що стосується алгоритмів, їх можна розділити на три типи.

Контрольоване навчання: це навчальний процес, у якому ви можете розглядати навчання під керівництвом вчителя. Це процес навчання алгоритму на

основі навчального набору даних. Він генерує функцію зіставлення між вхідною та вихідною змінною. Як тільки модель навчена, вона може почати робити прогнози/рішення, коли їй передаються нові дані. Декілька алгоритмів, які потрапляють у контрольоване навчання: лінійна регресія, логістична регресія, дерево рішень тощо.

Навчання без вчителя: це процес, при якому модель навчається з використанням інформації, яка не позначена. Цей процес можна використовувати для кластеризації вхідних даних до класів на основі їх статистичних властивостей. Це зазвичай називається кластерним аналізом, який означає угруповання об'єктів на основі інформації, знайденої в даних, що описує об'єкти або їх взаємозв'язок. Тут ціль полягає в тому, щоб об'єкти в одній групі були схожі один на одного, але відрізнялися від об'єктів в іншій групі. Небагато алгоритмів, які підпадають під навчання без вчителя, включають кластеризацію K-середніх, ієрархічну кластеризацію тощо.

Навчання з підкріпленням: слід концепції «вдарити та випробувати». Це навчання, взаємодіючи з простором чи довкіллям. Агент RL навчається на наслідки своїх дій, а не на явному навчанні. Це здатність агента взаємодіяти з навколишнім середовищем та визначати найкращий результат.

Саме для розробки рекомендаційної системи для підбору фільмів було використано контрольоване навчання.

### **Java Machine Learning Library (Java-ML)**

Java-ML – це відкритий Java фреймворк, який містить цілий набір різноманітних алгоритмів машинного навчання для програмістів. На сайті проекту можна знайти офіційну документацію з туторіалами щодо можливостей застосування Java-ML з прикладами коду [21].

JavaML – це набір алгоритмів машинного навчання, який має спільний інтерфейс для кожного типу алгоритму. Має хорошу документацію зі зрозумілими інтерфейсами. Ви також можете зібрати безліч кодів та посібників, призначених

для розробників програмного забезпечення чи програмістів. Деякі з його особливостей:

1. Маніпуляція даними;
2. Кластеризація;
3. Класифікація;
4. Бази даних;
5. Вибір функції;
6. Документація та ін.

### **Weka**

Weka це колекція алгоритмів машинного навчання, які можна застосовувати безпосередньо до датасету через наданий графічний інтерфейс або викликати через API [22].

Weka – це безкоштовна, проста бібліотека машинного навчання з відкритим вихідним кодом. Його назва нав'язана нелітаючим птахом, що мешкає на островах Нової Зеландії. Weka – це набір алгоритмів машинного навчання, який також підтримує глибоке навчання. Основна увага приділяється:

- збір даних;
- інструменти для підготовки даних;
- класифікація;
- регрес;
- кластеризація;
- візуалізація та ін.

### **RapidMiner**

RapidMiner - це платформа для обробки даних, що надає різні алгоритми машинного навчання через GUI та Java API. У Мережі можна знайти безліч посібників та інших матеріалів про застосування RapidMiner [23].

### **Генетичні алгоритми**

Генетичний алгоритм - це евристичний алгоритм пошуку, який використовується для вирішення задач оптимізації та моделювання шляхом випадкового підбору, комбінування та варіації шуканих параметрів з використанням механізмів, аналогічних до природного відбору в природі.

### **Jenetics**

Jenetics - це просунута бібліотека генетичних та еволюційних алгоритмів, генетичного програмування та багатоцільової оптимізації, написана на сучасному Java. На сайті проекту можна знайти документацію та туторіали для нових користувачів [24].

### **ESJ 23**

ESJ 23 - це заснований на Java дослідницький фреймворк із сильною підтримкою генетичних алгоритмів. Розроблено в лабораторії еволюційних обчислень ECLab Університету Джорджа Мейсона у США.

### **Watchmaker Framework**

Watchmaker Framework - це фреймворк, призначений для реалізації платформи-незалежних генетичних та еволюційних алгоритмів Java. На сайті проекту можна знайти інформацію про фреймворку з детальною документацією та прикладами його застосування [25].

## **Висновки до розділу 2**

В даному розділі було розглянуто методи створення рекомендаційної системи, а також розглянуто використання Java в розробці системи та використання штучного інтелекту з машинним навчанням у Java. Було виконано детальний аналіз кожного з них відносно виконання поставленої задачі. В ході аналізу було виявлено, що метод написання системи з нуля краще відповідає вимогам завдання. Для реалізації рекомендаційної системи було обрано мову програмування Java.

## 3 ЗАСОБИ РОЗРОБКИ ДЛЯ РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

### 3.1 Архітектура рекомендаційної системи

Зазвичай система пропонує користувачеві через системний інтерфейс надавати рейтинги для елементів для побудови та вдосконалення своєї моделі. Точність рекомендацій залежить від кількості оцінок, наданих користувачем. Єдиним недоліком цього методу є те, що він вимагає зусиль від користувачів, а також користувачі не завжди готові надати достатньо інформації. Незважаючи на те, що явний зворотний зв'язок вимагає більше зусиль від користувача, він все одно розглядається як надання більш надійних даних, оскільки він не передбачає вилучення переваг із дій, а також забезпечує прозорість процесу рекомендацій, що призводить до дещо вищої якості сприйняття рекомендацій і більше впевненості в рекомендаціях. Прогноз якісної рекомендації вбачає відгук користувача.

Фазу процесу оновлення інформації бази фільмів користувача зображено на рисунку 3.1. Система автоматично визначає уподобання користувача, відстежуючи різні дії користувачів, такі як оцінка фільмів та вибір жанру.

Неявний зворотний зв'язок зменшує навантаження на користувачів, роблячи висновки про уподобання їх користувачів з їх поведінки в системі. Метод хоч і не вимагає зусиль від користувача, але він менш точний. Крім того, також стверджувалося, що неявні дані переваг насправді можуть бути більш об'єктивними, оскільки немає упередженості, що виникає у користувачів, що реагують соціально бажаним способом, а також не виникає проблем із самозображенням або будь-якої потреби в підтримці іміджу для інших.

Система рекомендує або передбачає, яким фільмам користувач може віддати перевагу. Це можна зробити безпосередньо на основі набору даних, зібраного на етапі збору інформації, який може бути на основі пам'яті або на основі моделі, або завдяки спостережуваній системою діяльності користувача виділяє етапи рекомендацій.

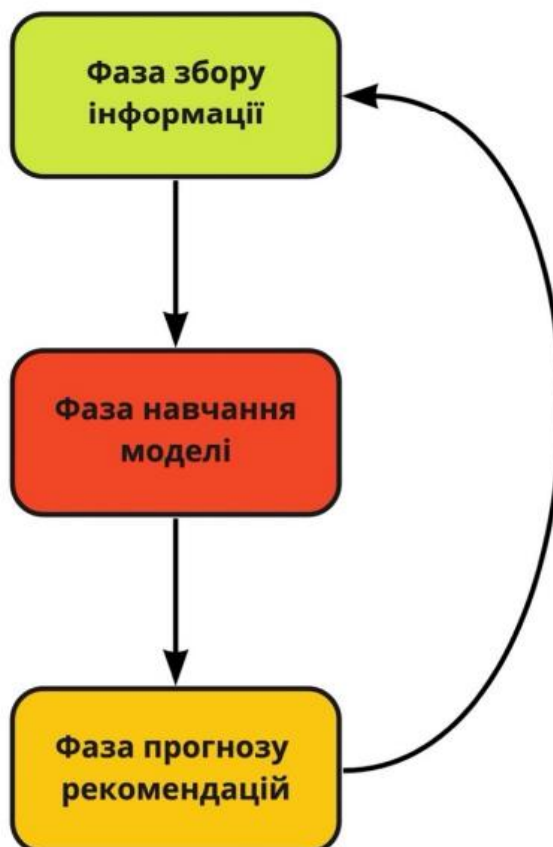


Рисунок 3.1 – Процес оновлення рекомендації користувачеві

Колаборативна фільтрація – це незалежна від домену техніка прогнозування вмісту, який неможливо легко та адекватно описати такими метаданими, як фільми та музика. Техніка спільної фільтрації працює шляхом побудови бази даних (матриця елементів користувача) переваг для елементів користувачами. Потім він відповідає користувачам з відповідними інтересами та уподобаннями, обчислюючи схожість між їхніми профілями, щоб робити рекомендації. Такі користувачі будують групу під назвою сусідство.

Користувач отримує рекомендації щодо тих елементів, які він раніше не оцінював, але які вже були позитивно оцінені користувачами в його сусідстві. Рекомендації, вироблені CF, можуть бути як передбаченнями, так і рекомендаціями.

Загалом рекомендаційна модель вбачає заздалегідь підготовлену базу користувача, котра з кожним новим входженням фільмів з оцінками буде



оновлювати прогнози моделі для користувача системою.

### 3.2 Середовище розробки IntelliJ IDEA

IntelliJ IDEA – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію. Сирцеві тексти Community-версії поширюються в рамках ліцензії Apache 2.0. Бінарні збірки підготовлені для Linux, Mac OS X і Windows.

Community версія середовища IntelliJ IDEA підтримує інструменти (у вигляді плагінів) для проведення тестування TestNG і JUnit, системи контролю версій CVS, Subversion, Mercurial і Git, засоби складання Maven, Ant, Gradle, мови програмування Java, Scala, Clojure, Groovy і Dart. Підтримується розробка застосунків для мобільної платформи Android. До складу входить модуль візуального проєктування GUI-інтерфейсу Swing UI Designer, XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту проєктів з Eclipse. Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Комерційна версія «Ultimate Edition» відрізняється наявністю підтримки додаткових мов програмування (наприклад, PHP, Ruby, Python, JavaScript, CoffeeScript, HTML, CSS, SQL), підтримкою технологій Java EE, UML-діаграм, підрахунок покриття коду, можливістю роботи з фреймворками (Rails, Grails, Google Web Toolkit, Spring, Play Framework і Hibernate), засобами інтеграції з Perforce, Microsoft Team Foundation Server і Rational ClearCase.

IntelliJ IDEA – інтегроване середовище розробки Java додатків від компанії JetBrains. Її позиціонують як найрозумніше та зручне середовище розробки для Java з підтримкою всіх останніх технологій та фреймворків.

IntelliJ IDEA входить до трійки найпопулярніших IDE для Java разом з Eclipse IDE та NetBeans IDE. Порівняння трьох середовищ розробки в даному матеріалі про вибір середовища для розробки.

Для створення програмного коду було обрано технології IntelliJ IDEA, а саме мова програмування Java.

Розроблений додаток містить форми для заповнення та кнопки для переходу.

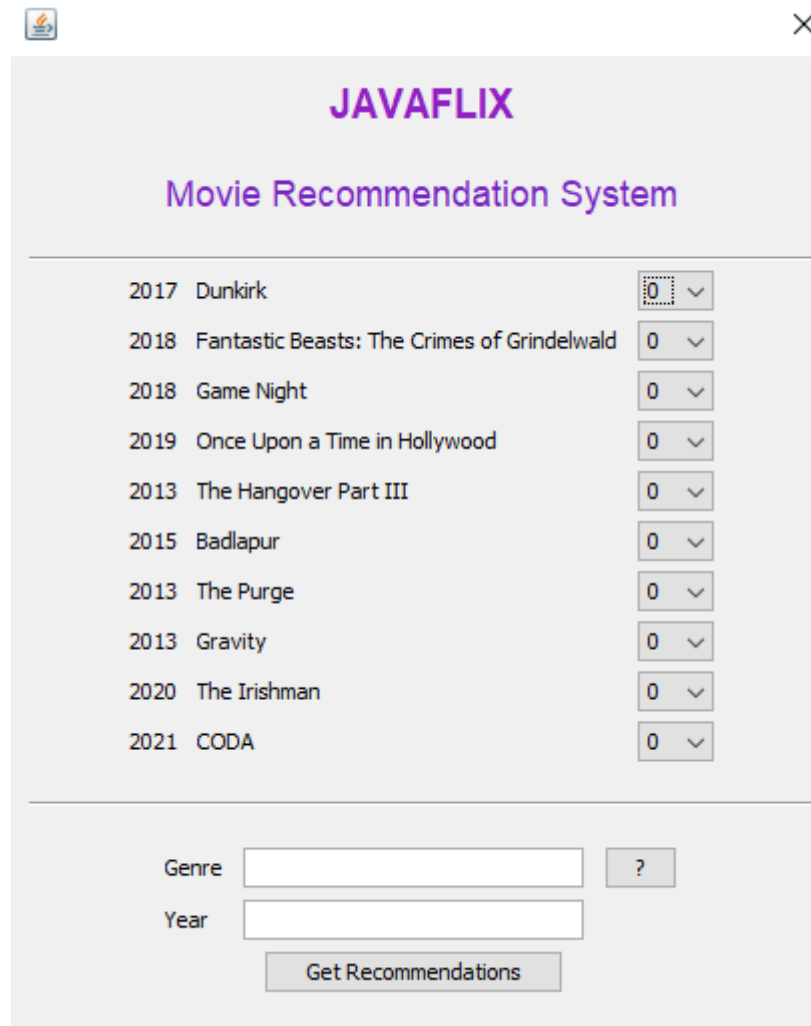
В ході розробки рекомендаційної системи були виконані такі пункти:

- постановка задачі побудови рекомендаційної системи та дослідження її типів;
- окреслення існуючих технологій для вирішення поставленої задачі;
- дослідження засобів розробки рекомендаційної системи;
- оцінка якості результатів;
- створення системи взаємодії з рекомендаційною системою.

### **3.3 Взаємодія користувача з рекомендаційною системою**

Коли користувач вперше використовує систему (т.з. холодний старт) система використовує штучно налаштовану базу фільмів, котра включає популярні фільми. Таке рішення дозволяє використовувати концепцію колаборативної фільтрації за наявних умов. Під час оновлення бази фільмів користувача модель оновлюється та будує рекомендації вже на основі нових оцінок.

Побудова прогнозів рекомендацій користувачеві відбувається завдяки його оцінкам, а також за введеними ним вподобаннями. Вподобання та оцінки користувача розташовані в інтерфейсі користувача (рисунок 3.2).



The screenshot displays the JAVAFLIX Movie Recommendation System interface. It features a list of movies with their years and titles, and a rating dropdown menu for each. Below the list are input fields for Genre and Year, and a 'Get Recommendations' button.

Year	Movie Title	Rating
2017	Dunkirk	0
2018	Fantastic Beasts: The Crimes of Grindelwald	0
2018	Game Night	0
2019	Once Upon a Time in Hollywood	0
2013	The Hangover Part III	0
2015	Badlapur	0
2013	The Purge	0
2013	Gravity	0
2020	The Irishman	0
2021	CODA	0

Genre  ?

Year

Рисунок 3.2 – Інтерфейс програми

Послідовність отримання прогнозу рекомендацій вбачає, що користувач натискає на оцінки (від 0 до 10), де 0 означає, що користувач оцінив фільм в 0 балів, а 10, відповідно, оцінив у 10 балів. Після того, як користувач зазначив кількість балів фільму, наступним кроком буде введення улюбленого жанру, потім введення бажаного року випуску фільму, а вже наступним кроком підтвердження вибору користувача, натиснувши кнопку «Get Recommendations», користувач запускає процес оновлення прогнозу рекомендацій (рисунок 3.3).

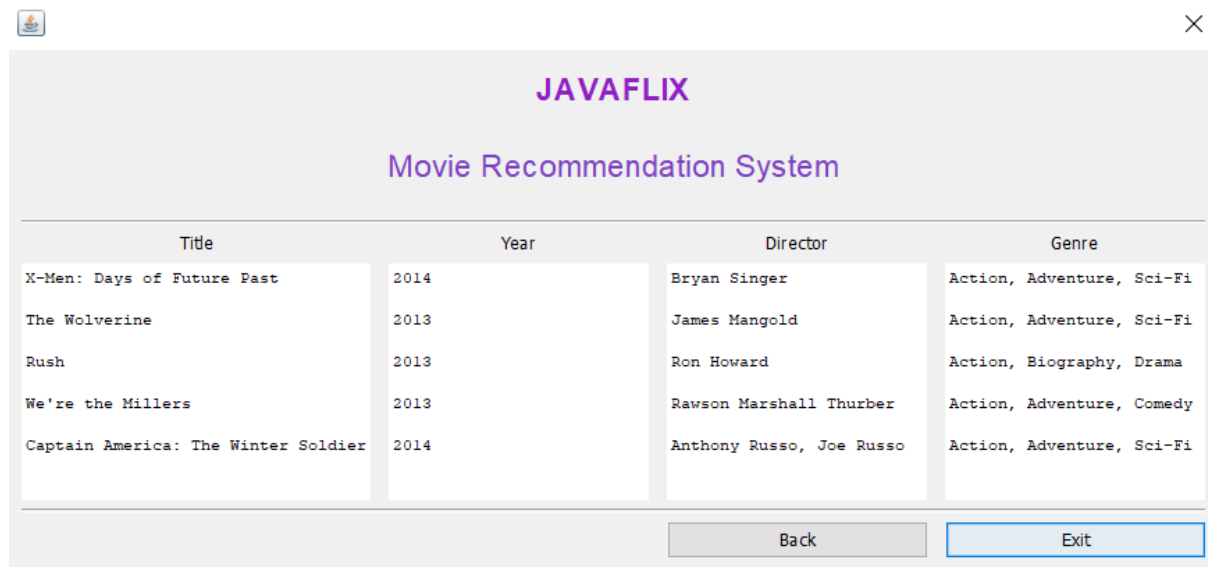


Рисунок 3.3 – Вікно виведення рекомендацій

Даний вигляд користувацького інтерфейсу, дозволяє швидко орієнтуватися в жанрах до фільмів, та дає можливість користувачеві корегувати виставлені оцінки до фільмів.

### 3.4 Засоби розробки інтерфейсу користувача

Від початку засоби Java створювалися як програмна платформа для створення так званих насичених інтернет-застосунків (rich internet applications), визначаючи архітектуру, каркас і стиль розробки застосунку. Оскільки Java надає велику кількість інтерфейсів і класів для розробки застосунків графічного інтерфейсу користувача, засоби JavaFX фактично є сучасною альтернативою бібліотеці `java.swing`. Перша версія платформи (JavaFX 1.0) вийшла у 2008 році та включала спеціальну скриптову мову JavaFX Script для опису графічного інтерфейсу. У 2011 році вийшла версія JavaFX 2.0 під орудою Oracle. Розробники цієї версії відмовилися від спеціальної скриптової мови на користь Java. Версія JavaFX 8, яка вийшла у 2014 році, розроблена у відповідності до можливостей та стилю Java 8, надає можливості роботи з 3D-графікою, а також пропонує нові візуальні компоненти. Номер версії відповідає Java 8, тому версії 3, 4, 5, 6 і 7 відсутні.

Основними рисами, що відрізняють JavaFX від попередніх бібліотек

підтримки графічного інтерфейсу користувача є такі:

- вбудована підтримка патерну проектування MVC (Model-View-Controller);
- можливість декларативного опису візуальних компонентів (мова FXML);
- сучасний стиль візуальних компонентів;
- підтримка розширених можливостей взаємодії користувача з застосунком;
- можливість використання CSS-стилів для стилізації елементів користувацького інтерфейсу;
- можливість використання 3D-графіки;
- спрощена модель розгортання застосунків.

Є також низка додаткових можливостей, пов'язаних з графікою, текстом, взаємодією з раніше створеними бібліотеками і т. д.

### **3.5 Оцінки точності для рекомендаційних алгоритмів**

Якість рекомендаційного алгоритму можна оцінити, використовуючи різні типи вимірювань, які можуть бути точністю чи охопленням. Тип використовуваних показників залежить від типу техніки фільтрування.

Точність – це частка правильних рекомендацій від загальної кількості можливих рекомендацій, тоді як охоплення вимірює частку об'єктів у просторі пошуку, для яких система може надати рекомендації. Метрики для вимірювання точності систем фільтрації рекомендацій поділяються на статистичні та метрики точності підтримки прийняття рішень. Придатність кожного показника залежить від особливостей набору даних та типу завдань, які буде виконувати система, що рекомендує. Статистичні показники точності оцінюють точність техніки фільтрації шляхом порівняння прогнозованих оцінок виставлених користувачем.

### **3.6 Обґрунтування вибору програмної реалізації**

Після детального аналізу предметної області та дослідження

рекомендаційних систем та їх розробки було вирішено розроблювати програмний комплекс на базі колаборативної фільтрації використовуючи мову програмування Java.

### **3.7 Вимоги до рекомендаційної системи**

Для інтеракції з рекомендаційною системою програмний продукт має відповідати наступним вимогам:

- інтерфейси користувача (User Interfaces) – система повинна дозволяти повну навігацію і вибір інтерактивних дій із використанням тільки миші;

Вимоги продуктивності (Performance Requirements):

- система повинна бути повністю завантажена не більше ніж за 5 секунд, а генерація прогнозу рекомендаційної системи користувача повинна відбуватися не більше ніж 2 секунди після того, як користувач відправить оцінку фільму;
- загальна швидкість побудови прогнозів (інференс) рекомендацій нейронною мережею повинно бути швидше за 10000 одиниць за 5 секунд.

Набір даних MovieLens, котрий містить 10136 рейтингів у 3144 фільмах. Ці дані були створені 610 користувачами між 29 березня 1980 року та 24 вересня 2021 року.

### **3.8 Вимоги до програмного забезпечення**

Системні вимоги:

- операційна система Microsoft Windows 7/8/10;
- клавіатура;
- миша;
- розміри вільного дискового простору 200 МБ;
- роздільна здатність екрана – не менше 1024×768 пікселів.

### **3.9 Проектування дизайну інтерфейсу користувача**

Головна функція, котру повинен виконувати дизайн рекомендаційної

системи – інтуїтивно зрозуміле користування. Саме тому запропоновано використання списку фільмів у вигляді блоків, де кожен блок підтримує додавання оцінок у вигляді шкали від 0 до 10. Відповідно до шкали оцінювання фільму та складання загального рейтингу фільмів користувачів будь-якого сервісу. Інтерфейс користувача системою зображено на рисунку 3.4.

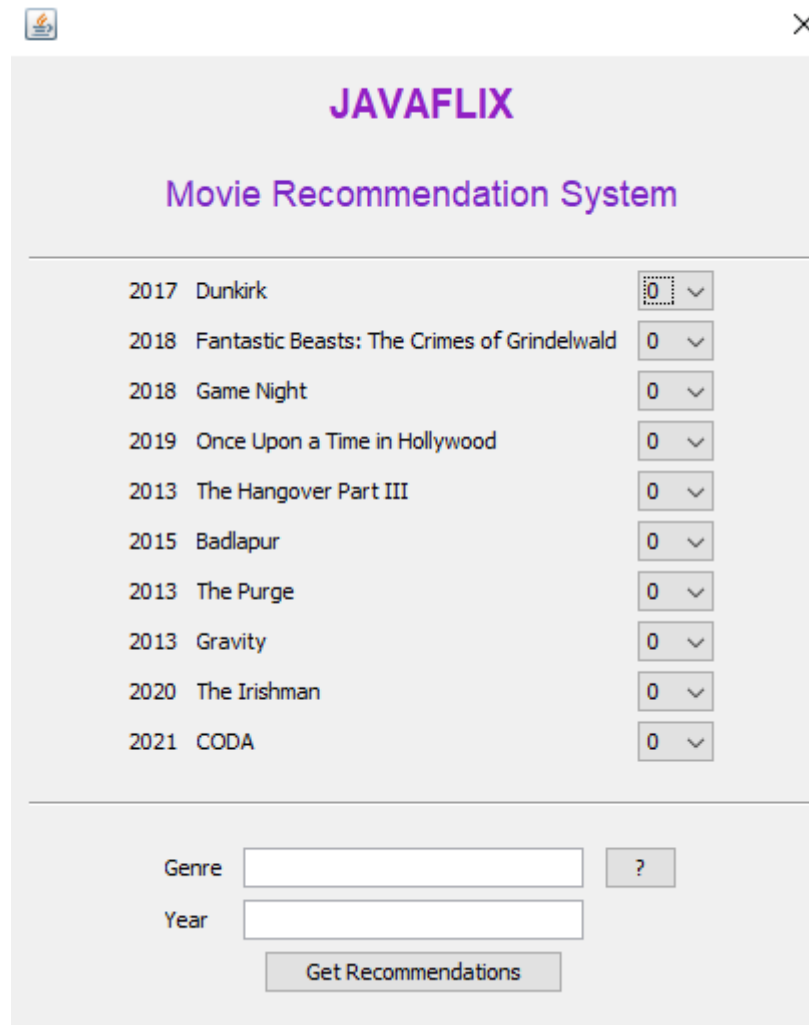


Рисунок 3.4 – Інтерфейс програми

### 3.10 Системні вимоги та інсталяція

Для створення системи було використано: мову програмування Java. Графічний користувальницький інтерфейс реалізовано з використанням інтерфейсу програмування додатків Java.ML.

Мова програмування Java є однією з найпотужніших та найпопулярніших мов, яка динамічно розвивається і є затребуваною в ІТ-галузі. Рекомендаційна

система потребує попередньо встановленого IntelliJ IDEA будь-якої версії, починаючи з 2013 року.

Для того, щоб скористатися системою, необхідно:

- відкрити IntelliJ IDEA;
- натиснути Open project or Open folder;
- обрати папку Movie recommendation system;
- натиснути клавішу Enter;
- знайти папку TempLoader та відкрити її;
- натиснути кнопку Run (рис.3.5).

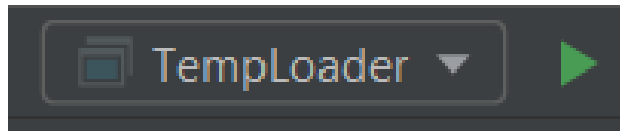


Рисунок 3.5 – Кнопка Run для запуску програми

### 3.11 Клієнтський інтерфейс

Коли користувач відкриває рекомендаційну систему, він має можливість переглянути назви з найпопулярніших фільмів з різними жанрами та оцінити їх, отримавши при цьому персоналізовані рекомендації. Для перегляду рекомендацій фільмів, користувач спочатку повинен оцінити фільм з запропонованого списку (рис.3.6).



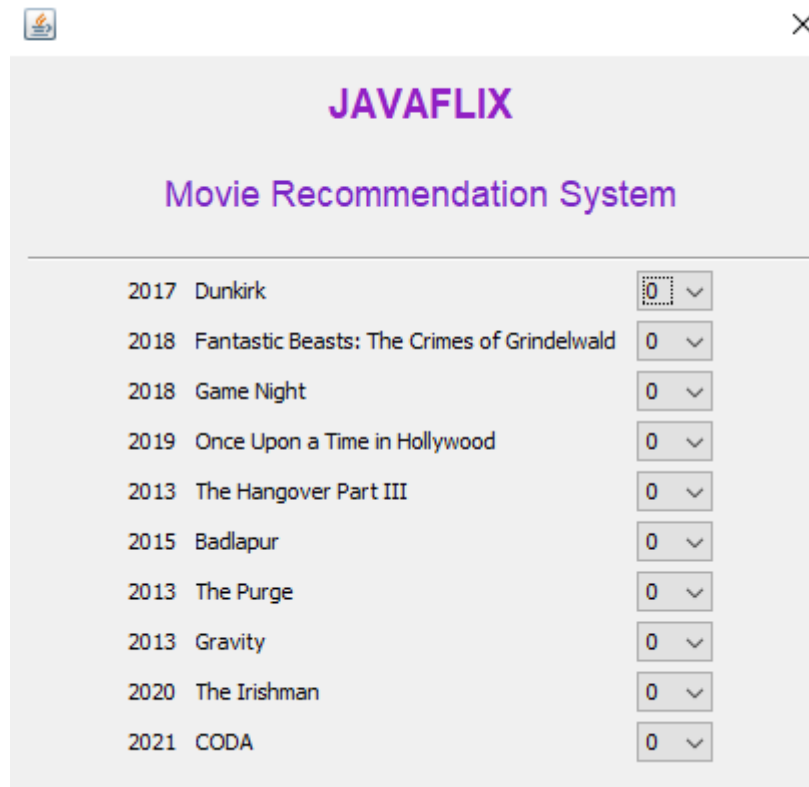


Рисунок 3.6 – Вікно програми для оцінки фільмів

Потім користувач повинен ввести жанр фільму, який він хоче подивитись та рік за який він хоче подивитись цей фільм. Наприклад, якщо користувач введе 2013 рік, то система йому буде пропонувати фільми починаючи з 2013 року та вище.

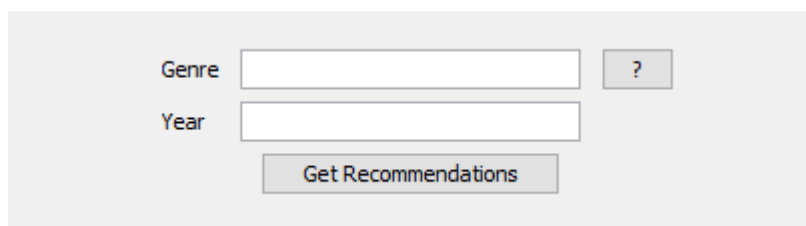
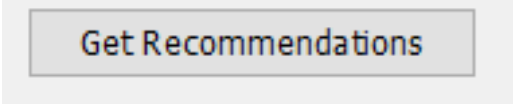


Рисунок 3.7 – Вікно програми для введення жанру та року

А вже потім користувач повинен натиснути кнопку «Get Recommendations», щоб отримати запропоновані йому системою рекомендації фільм за його вподобань.



Get Recommendations

Рисунок 3.8 – Кнопка для отримання рекомендацій

### **Висновки до розділу 3**

В цьому розділі було проведено дослідження засобів розробки рекомендаційних систем та нейронних мереж. Було розглянуто архітектуру рекомендаційної системи, ознайомлено з середовищем розробки, проведено оцінки точності для рекомендаційних систем, Проведено обґрунтування вибору програмної реалізації та засобів створення рекомендаційної системи з інтерфейсом користувача, а також вказано вимоги до рекомендаційної системи та програмного забезпечення.

**Спеціальний розділ**

**ОХОРОНА ПРАЦІ**

**до бакалаврської кваліфікаційної роботи**

на тему:

**«РЕКОМЕНДАЦІЙНА СИСТЕМА З ПІДБОРУ ФІЛЬМІВ ЗА  
ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ»**

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 401.21810111**

*Виконала студентка 4го курсу, групи 401*

\_\_\_\_\_ *М. Р. Бuzінська*

«\_\_\_» червня 2022 р.

*Консультант ст. викл.*

\_\_\_\_\_ *О. В. Макарова*

«\_\_\_» червня 2022 р.

## 4 СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ

### 4.1 Вступ

Охорона праці – це система законодавчих, організаційно-технічних, соціально-економічних, санітарно-гігієнічних і лікувально-профілактичних мір і засобів, спрямованих на збереження життя, здоров'я й працездатності людини в процесі праці.

В даній кваліфікаційній роботі створено рекомендаційну систему з підбору фільмів. З метою створення безпечних і здорових умов праці при створенні цієї програми виконується аналіз умов на робочому місці.

**Метою роботи** є створення безпечних і здорових умов праці на робочих місцях, в робочих зонах, у виробничих приміщеннях та забезпечення безпеки людини у надзвичайних ситуаціях.

**Завдання** охорони праці полягає в тому, щоб звести до мінімуму ймовірність поразки працюючого під дією небезпечного виробничого фактора або захворювання під дією шкідливого виробничого фактора з одночасним забезпеченням комфортних умов при максимальній продуктивності праці. Закон України "Про охорону праці" визначає основні положення по реалізації конституційного права громадян на охорону їх життя і здоров'я в процесі трудової діяльності; регулює взаємини між адміністрацією і працівником в незалежності від форм власності; встановлює єдиний порядок організації охорони праці в Україні.

Забезпечення комфортних і безпечних умов праці є важливим обов'язком роботодавця. Навколишнє середовище, у якому працює людина, безпосередньо впливає на її здоров'я, самопочуття і, як наслідок, на її працездатність і продуктивність.

## 4.2 Опис обраного виробничого приміщення, робочих місць, їх обладнання та складання вихідних даних для кількісної оцінки умов праці

Приміщення розташовано на другому поверсі двоповерхової будівлі, що знаходиться у м. Миколаєві на Богоявленському проспекті, 42А. Розміри приміщення складають  $a * b * c = 6,5 * 3,5 * 2,5$ м. У приміщенні присутні три металопластикових вікна (з подвійним склопакетом) розмірами  $c * d = 0,7 * 1,3$ м.

Приміщення має сучасний офісний інтер'єр. Стеля виконана у вигляді підвісної конструкції із синтетичного матеріалу світло-сірого кольору. Стіни мають гладку поверхню кольору білого кольору. Підлога має покриття із лінолеуму, що імітує паркет світло-коричневого кольору.

Вікна обладнані світлозахисними пристроями у вигляді вертикальних регульованих жалюзі. У приміщенні розташовано 4 робочі місця, обладнані сучасними персональними комп'ютерами з необхідними периферійними пристроями, один лазерний принтер, 4 телефони. Для зберігання робочої документації та науково-технічної літератури передбачена шафа.

Загальний вигляд обраного виробничого приміщення представлено на рис. 4.1.

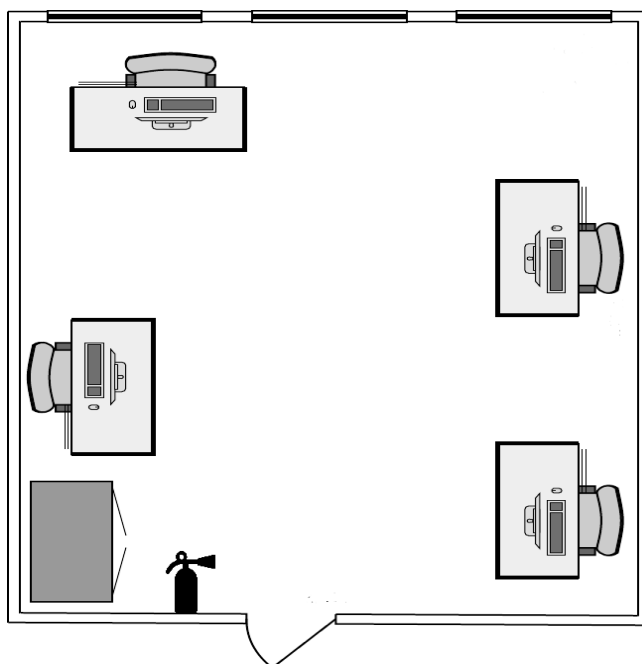


Рисунок 4.1 – Загальний вигляд обраного виробничого об'єкта

Напруга джерела живлення електроспоживної техніки – 220 В з глухо

заземленою нейтралю, з частотою 50 Гц (згідно НПАОП 0.00-1.28-2010) [3].

Електромережа виконана у вигляді трипровідної з дотриманням усіх вимог нормативних документів.

Таблиця 4.1 - Фактичні та нормативні значення параметрів приміщення.

Параметр	Норма	Фактичні параметри
Площа, S	Не менше ніж 6 м <sup>2</sup>	8,75 м <sup>2</sup>
Об'єм, V	Не менше ніж 20 м <sup>3</sup>	23,6 м <sup>3</sup>

Показники відповідають вимогам нормативних документів.

Мікрокліматичні умови у літній період забезпечується системою кондиціонування, потужності якої вистачає для забезпечення комфортних умов праці. У зимовий період опалення здійснюється центральною системою, яка забезпечує необхідний тепловий режим відповідно до вимоги ДБН В.2.5-67:2013 [2].

Таблиця 4.2 - Значення мікроклімату

Період року	Параметр	Оптимальний	Фактичний
Теплий	Температура	23 – 25	28
	Вологість	60 – 40	50
	Швидкість повітря	< 0,1 м/с	
Холодний	Температура	22 – 24	22
	Вологість	60 – 40	40
	Швидкість повітря	< 0,1 м/с	

Значення відносної вологості повітря в холодний період року знаходиться на межі допустимих значень, доцільно використовувати в цей період зволожувачі повітря.

Для забезпечення нормального освітлення застосовуються природне, однібічне, бічне і штучне освітлення, а також сполучене, які нормуються

санітарними нормами й правилами ДБН В.2.5-28-2006 [11].

Пожежна безпека в обраному виробничому приміщенні забезпечується дотриманням вимог НПАОП 0.00-1.28-10 [4].

У приміщенні технічного відділу причинами шуму і вібрації є апарати, і прилади устаткування: друкуючі пристрої, комп'ютери, кондиціонер та ін.

За даними контрольних обстежень, виконанням необхідних вимірів, а також експертних оцінок здійснена оцінка умов праці (основних факторів виробничого середовища та трудового процесу) в обраному виробничому приміщенні (табл. 4.3). Наведені у табл. 4.3 відомості є вихідними даними для виконання подальших розрахункових робіт даного розділу дипломної роботи бакалавра.

Таблиця 4.3 – Елементи умов праці для відділу ГП НПКГ «Зоря»- «Машпроект»

№ з / п	Фактори умов праці на робочому місці		Значення індикатора	Тривалість дії фактора, хв.
1	Температура на робочому місці (PM) У виробничій кімнаті:°C -теплий період -холодний період		25	480
			23	480
2	Відносна вологість в RM,%		50	480
3	Швидкість вітру в RM, м / с		0,2	420
4	RM, Lux освітлення		200	300
5	Мінімальний розмір об'єкта Впізнаваність, мм		0,5	300
6	Виробничий шум, дБА		0,9	480
7	термічна міцність Копія, Вт / м <sup>2</sup>		50	240
8	Токсичні речовини, Озон, більше, ніж різноманітність ГДК		-	-
9	Промислова пил (папір і ін.), Різноманітність за межами ГДК		0,6	480
10	Робоче місце (RM), зовнішнє і просторове переміщення		Робоче місце стаціонарне, маса переміщення вантажу до 5 кг	480
11	Кількість важливих об'єктів спостереження		3	480
12	період концентрації Спостереження,% часу зміни		55	360
13	період повторення Робота, п		25	360
14	різноманітність робіт		Ранкова	480
15	безперервний період День, час		8	420
16	Режим роботи і відпочинку		Відсутність обґрунтованого режиму праці та відпочинку	480
17	Напруга і емоційне навантаження		Складні дії за заданим планом з можливістю корегування	300
18	Кількість рухів пальців на годину		1080	360
19	Одноманітність, тривалість роботи, Повторення, р		25	360



### 4.3 Інтегральна оцінка умов праці в обраному виробничому приміщенні

Для інтегральної оцінки умов праці [5] в обраному виробничому приміщенні слід скористатися даними табл. 4.3 та здійснити оцінку питомої ваги кожного із представлених там факторів виробничого середовища та трудового процесу, скориставшись наведеними у додатку А даного розділу роботи критеріями бальної оцінки.

У табл. 4.4 представлені параметри, що необхідні для інтегральної оцінки умов праці:

$x_{ni}$  – нормативне значення  $i$  – того фактору умов праці (прийняті значення відповідають оптимальному (допустимому) класу умов праці згідноз Гігієнічною класифікацією);

$x_{bi}$  – дійсне значення  $i$  – того фактору умов праці (відповідно до даних табл. 4.3);

$x_{xi}$  – оцінка  $i$  – того фактору умов праці (відповідно до даних додатку А даного розділу роботи), балів;

$t_i$  – тривалість дії  $i$  – того фактору умов праці (відповідно до даних табл. 4.3), хв.;

$t_{pit_i}$  – відносна тривалість дії  $i$  – того фактору умов праці (за прийнятої тривалості робочої зміни  $t_p = 480$  хв.), хв., тобто:

$$t_{num_i} = \frac{t_i}{t_p} = \frac{t_i}{480}; \quad (4.1)$$

$x_{\phi_i}$  – фактична оцінка питомої ваги  $i$  – того фактору умов праці, балів, а саме:

$$x_{\phi_i} = x_{x_i} t_{num_i} = x_{x_i} \frac{t_i}{480}. \quad (4.2)$$

За даними табл. 4.4 визначаємо елемент умов праці, який одержав у балах найбільшу оцінку.

Таблиця 4.4 Параметри, необхідні для розрахунку комплексної бальної оцінки умов праці на робочому місці

№з / п	Фактор умов праці на робочому місці	Нормоване значення фактора $X_{нi}$	Оцінка фактора		Тривалість дії фактора		Фактична оцінка питомої ваги фактора $X_{fi}$
			Абсолютна $X_{abi}$	У балах $X_{xi}$	Хвилин $t_i$	У долях робочої зміни $t_{пит i}$	
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С - теплий період - холодний період	23...25	25	1	480	1	1
		21...23	23	1	480	1	1
2	Відносна вологість повітря на РМ, %	40..60	50	1	480	1	1
3	Швидкість руху повітря на РМ, м/с	<0,2	0,2	2	420	0,875	1,75
4	Освітленість на РМ, лк	400	200	3	300	0,625	1,85
5	Мінімальний розмір об'єкта розпізнавання, мм	>1	0,5	2	300	0,625	1,25
6	Виробничий шум, дБА	< 1	0,9	1	480	1	1
7	Інтенсивність теплового випромінювання, Вт/м <sup>2</sup>	≤140	50	2	240	0,5	1
8	Токсична речовина, озон, кратність перевищення ГДК	≤1	-	1	480	1	1
9	Виробничий пил ( паперовий), кратність перевищення ГДК	≤1	0,6	2	480	1	2
10	Робоче місце (РМ), поза та переміщення у просторі	РМ стаціонарне, маса переміщення до 5 кг	Робоче місце стаціонарне, маса переміщення вантажу до 5 кг	1	480	1	1

Кафедра інтелектуальних інформаційних систем  
Рекомендаційна система з підбору фільмів за допомогою методів машинного навчання

Закінч. табл. 4.4

11	Кількість важливих об'єктів спостереження	<5	3	1	480	1	1
12	Тривалість зосередженого спостереження, % часу зміни	<25	55	3	360	0,75	2,25
13	Тривалість повторюваних операцій, с	>100	25	3	360	0,75	2,25
14	Змінність роботи	Ранкова	Ранкова	1	480	1	1
15	Тривалість безперервної роботи за добу, годин	<8	8	2	420	0,875	1,75
16	Режим праці та відпочинку	Обґрунтований з вкл.. музики та гімнастики	Відсутність обґрунтованого режиму праці та відпочинку	3	480	1	3
17	Нервово-емоційне навантаження	Прості дії за індивідуальним планом	Складні дії за заданим планом з можливістю корегування	3	300	0,625	1,875
18	Кількість рухів пальців на годину	<360	1080	3	360	0,75	2,25
19	Монотонність, тривалість операцій, які повторюються, с	>100	25	3	360	0,75	2,25

Таким елементом є елемент  $x_{15}$ , який пов'язаний з тривалістю роботи за добу, тобто  $x_{\max}=x_2=3$ . Даний елемент вважається визначаючим.

Далі розраховується:

1. Середній бал усіх елементів крім визначаючого  $\bar{x}$ , балів:

$$\bar{x} = \frac{\sum_{i=1}^{n-1} x_{\phi i}}{n-1}, \quad (4.3)$$

де  $n$  – фактична кількість врахованих елементів умов праці (у даному випадку  $n = 19$ ).

Тоді:

$$\bar{x} = \frac{\sum_{i=1}^{n-1} x_{\phi i}}{n-1} = \frac{1+1+1+1,75+1,875+1,25+1+1+1+2+1+1+2,25+2,25+1+1,75+3+1,875+2,25+2,25}{19-1} = 1.75$$

2. Інтегральна бальна оцінка умов праці на робочому місці у відділі  $I_n$ , балів.

$$I_n = 10 * \left( x_{\max} + \frac{6 - x_{\max}}{6} \right) = 10 * \left( 3 + \frac{1,75 * 6 - 3}{6} \right) = 38,75$$

Отримане значення інтегральної оцінки умов праці порівнюємо зі значеннями, наведеними в додатку Б даного розділу магістерської наукової роботи і робимо висновок відносно категорії важкості праці.

Згідно даними додатку Б умови праці на визначеному робочому місці [5] відносяться до III категорії, коли спостерігається відхилення від ГДК і ГДР факторів трудового процесу та допустимих величин психофізіологічних факторів.

#### 4.4 Оцінка ефективності заходів щодо покращення умов праці

Пропонується до всіх факторів умов праці, бальна оцінка яких перевищує значення  $x_{x_i} = 2$ , вжити заходи (надати рекомендації) з метою досягнення кожним із розглянутих елементів умов праці саме зазначеного вище значення ( $x_x = 2$ ).

За даними табл. 5.2 визначаємо інтегральний показник важкості праці за формулою, що використовується, коли умови праці оцінюються балами «1» або «2»:

$$I_{n_2} = 19,7 \times \bar{x} - 1,6 \times \bar{x}^2,$$

де

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n}$$

Тоді для даних умов праці згідно табл. 6.2 маємо:

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n} = \frac{1 + 1 + 1 + 1,75 + 1,875 + 1,25 + 1 + 1 + 1 + 1 + 1 + 1 + 1,75 + 1,875 + 6 \times 2}{19} = 1,55;$$

$$I_{n_2} = 19,7 \times \bar{x} - 1,6 \times \bar{x}^2 = 19,7 \times 1,55 - 1,6 \times 1,55^2 = 26,73 \text{ балів}$$

Відповідно до даних, наведених у додатку Б, отримане значення інтегрального показника  $I_{n_2} = 26,73$  балів, відповідає II категорії умов праці, тобто роботам, що виконуються в умовах, які відповідають гранично допустимим концентраціям (ГДК) і рівням (ГДР) санітарно-гігієнічних елементів праці, а також допустимим величинам психофізіологічних факторів.

Інтегральний показник важкості праці дозволяє визначити вплив умов праці на працездатність людини у такій послідовності:

1. Ступінь втоми працівника на визначеному робочому місці  $B$ , у. о.

$$B = \frac{I_n - 15,6}{0,64},$$

де чисельні значення 15,6 і 0,64 – це коефіцієнти регресії.

Тоді:

– до впровадження комплексу заходів з охорони праці коефіцієнт втоми складає

$$B_1 = \frac{I_n - 15,6}{0,64} = \frac{38,75 - 15,6}{0,64} = 36,17;$$

– після впровадження комплексу заходів з охорони праці коефіцієнт втоми складає

$$B_2 = \frac{I_{n_2} - 15,6}{0,64} = \frac{26,73 - 15,6}{0,64} = 17,39.$$

2. Рівень працездатності людини  $\Pi$ , у. о.

$$\Pi = 100 - B$$

Тоді:

– до впровадження комплексу заходів з охорони праці рівень працездатності складає

$$\Pi_1 = 100 - B_1 = 100 - 36,17 = 63,8;$$

– після впровадження комплексу заходів з охорони праці рівень працездатності складає

$$P_2 = 100 - B_2 = 100 - 17,39 = 82,6.$$

3. Зміна продуктивності праці  $\Delta P$ , %.

$$\Delta P = 0,2 \cdot \left( \frac{P_2}{P_1} - 1 \right) \cdot 100 = 0,2 \cdot \left( \frac{82,6}{63,8} - 1 \right) \cdot 100 = 5,9\%$$

Виконані розрахунки довели, що проведені заходи з охорони праці призвели до зменшення важкості праці з III до II категорії, і відповідно, зниженню втоми, підвищенню працездатності працівників.

Крім того, вказані заходи можуть призвести до підвищення продуктивності праці 5,9 %.

### **Техніка безпеки**

Згідно з нормативним документом НПАОП 40.1-1.01-97 Правила безпечної експлуатації електроустановок (ДНАОП 1.1.10-1.01-97) [4], всі виробничі маніпуляції на працюючих електроустановках повинні оформлятися офіційним завданням на стандартному бланку (т.зв. наряд-допуск), в якому зазначаються:

- зміст і місце проведення робіт;
- початок і кінець маніпуляцій;
- умови електробезпеки для їх здійснення;
- список працівників;
- прізвище відповідального за безпеку.

Також можливе здійснення таких робіт за розпорядженням керівництва і на підставі списку маніпуляцій, необхідних для поточного функціонування електроустановок. Проводити маніпуляції без підписаного наряду-допуску

забороняється, не можна також самовільно збільшувати обсяг робіт або складу бригади. Якщо необхідно провести капітальний ремонт електричних агрегатів з напругою понад 1 тис. Вольт без виключення електроенергії або повітряних ліній електропередач, всі роботи повинні здійснюватися відповідно до технологічних карт, підписаними директором підприємства. Порядок робіт на електроустановках з 1 тис. Вольт і вище напругою наступний:

- вимикається напруга з струмопровідних ділянок, які перебувають біля робочого поля (або ставиться огорожу) для виключення випадкового торкання;
- надаватися діелектричне взуття, або використовується електроізолюючими килимок;
- готується інструмент з ізоляцією (наприклад, для викруток необхідно забезпечити електроізоляцію стрижня), або надягають діелектричні рукавички;
- якщо роботи проводяться в приміщеннях з електроприладами, в підземних колодязях, різних тунелях і т.п., надягають захисні каски.

### **Гігієна праці і виробнича санітарія**

Виходячи з принципів Гігієнічної класифікації праці, умови розподіляють на 4 класи:

1) клас - оптимальні умови праці - такі умови, при яких зберігається не лише здоров'я працюючих, а й створюються передумови для підтримання високого рівня працездатності;

2) клас - допустимі умови праці - характеризуються такими рівнями факторів виробничого середовища і трудового процесу, які не перевищують встановлених гігієнічних нормативів, а можливі зміни функціонування стану організму відновлюються за час регламентованого відпочинку або до початку наступної зміни та не справляють негативного впливу на стан здоров'я;

3) клас - шкідливі умови праці - характеризуються такими рівнями



шкідливих виробничих факторів, які перевищують гігієнічні нормативи і здатні чинити несприятливий вплив на організм. Шкідливі умови праці за ступенем перевищення гігієнічних нормативів та вираженості можливих змін в організмі людини поділяють на чотири ступені: перший ступінь характеризується рівнями шкідливих виробничих факторів, які викликають функціональні зміни, що виходять за межі фізіологічних коливань та збільшують ризик погіршення здоров'я; другий ступінь - це умови праці, які характеризуються рівнями шкідливих факторів виробничого середовища та трудового процесу, які викликають функціональні зміни, що виходять за межі фізіологічних коливань та збільшують ризик погіршення здоров'я; третій ступінь - умови праці, які характеризуються рівнями шкідливих факторів виробничого середовища і трудового процесу, які призводять до зростання виробничо обумовленої захворюваності, до розвитку професійних захворювань; четвертий ступінь - умови праці характеризуються рівнями шкідливих факторів виробничого середовища і трудового процесу, які здатні призводити до значного зростання хронічної патології та рівнів захворюваності з тимчасовою втратою працездатності, розвитку важких форм професійних захворювань (з втратою загальної працездатності);

4) клас - небезпечні (екстремальні) умови праці - характеризуються такими рівнями шкідливих факторів виробничого середовища і праці, вплив яких протягом робочої зміни створює загрозу для життя, високий ризик виникнення важких форм гострих професійних уражень.

Мікроклімат приміщень - умови внутрішнього середовища цих приміщень, що впливають на тепловий обмін працюючих з оточенням шляхом конвекції, кондукції, теплового випромінювання та випаровування вологи відповідно до ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень (табл.4.4) [4].

Таблиця 4.5 - Класифікація робіт за важкістю та енерговитратами (ДСН 3.3.6.042-99)

Класифікація робіт	Характеристика робіт	Енерговитрати, Вт, (ккал/год)
I-а - легка	Виконуються сидячи і не потребують фізичного напруження	105- 140 (90-120)
I-б - легка	Виконуються сидячи, стоячи або в русі з незначними фізичними навантаженнями	141 - 175 (121-150)
II-а - середньої важкості	Виконуються в русі при переміщенні вантажів вагою до 1 кг або сидячи чи стоячи з фізичними навантаженнями	176-232(151-200)
II- важка	Пов'язані з хворобою, переміщенням та перенесенням вантажів вагою до 10 кг і супроводжуються помірним фізичним навантаженням	233-290(201-250)
III - важка	Постійне переміщення з перенесенням вантажів (понад 10 кг), що потребують значних фізичних зусиль	291 -349(251-300)

Значення параметрів мікроклімату суттєво впливають на самопочуття працездатність людини. Тривала дія високої температури тіла людини при одночасній підвищеній його вологості призводить до збільшення температури людини до 38-40 град, (гіпертермія). При підвищенні температури значно збільшується потовиділення, внаслідок чого настає різке порушення водного обміну.

З потом із організму виділяється значна кількість солей, головним чином хлористого натрію, калію, кальцію. Зростає вміст у крові молочної кислоти, сечовини, внаслідок чого вона згущується. Перегрів тіла людини супроводжується головними болями, запамороченням, нудотою, загальною слабкістю, часом можуть виникати судоми та втрата свідомості. Негативна дія високої температури збільшується при підвищеній вологості.

Суттєві фізіологічні зміни в організмі можливі також при низькій температурі, яка призводить до переохолодження організму (гіпотермія). Найбільш вираженими реакціями на низьку температуру є звуження судин м'язів та шкіри. Охолодження тіла викликає порушення рефлексорних реакцій, зниження тактильних та інших реакцій, утруднення рухів. Все вищевказане може бути причиною травматизму.

Недостатня вологість повітря (нижче 20%) призводить до підсихання слизових оболонок дихальних шляхів та очей, внаслідок чого зменшується захисна здатність протистояти мікробам.

#### **Висновки до розділу 4**

У даному розділі роботи було розглянуто питання охорони праці на підприємстві ГП НПКГ «Зоря»-«Машпроект», виконана інтегральна оцінка умов праці, проведені заходи, спрямовані на їх покращення.

За допомогою інтегральної оцінки було виявлено наступні проблеми:

- недостатній рівень освітленості на робочому місці;
- високий рівень тривалості зосередженого спостереження;
- мала тривалість повторюваних операцій;
- відсутність обґрунтованого режиму праці та відпочинку;
- нервово-емоційне навантаження складними діями за заданим

планом з можливістю корегування;

- завелика кількість рухів пальців за годину;
- замала тривалість операцій, монотонність, які повторюються;

для покращення роботи підприємства були проведені наступні заходи:

- додано освітлення на робочому місці;
- зменшено тривалість зосередженого спостереження;
- збільшено тривалість повторюваних операцій;

- змінено режим праці та відпочинку на обґрунтований, без

включення музики та гімнастики;

## ВИСНОВКИ

Під час виконання даної роботи було розроблено рекомендаційну систему з підбору фільмів за допомогою методів машинного навчання.

Проаналізовано актуальність проблеми підбору фільмів та її важливість у наш час. Були розглянуті аналоги відомих сайтів, що присвячені підбору фільмів. Проведено дослідження різноманітних уявлень та тлумачень про рекомендаційні системи з підбору фільмів. Надано можливість порівняння різних точок зору користувачів, а також максимально зручно донесено необхідну інформацію. Було проаналізовано предметну сферу для більш чіткого розуміння як краще реалізовувати систему.

В ході виконання кваліфікаційної роботи було розглянуто різні шляхи створення систем, проаналізовано наявні аналоги та зібрано інформацію про різні шляхи створення систем. Було описано особливості їх використання, переваги та недоліки.

Рекомендаційні системи є найпопулярнішим типом програм машинного навчання, які використовуються у всіх секторах. Вони є вдосконаленням порівняно з традиційними алгоритмами класифікації, оскільки вони можуть приймати багато класів введення та надавати алгоритми на основі рейтингу подібності, щоб надати користувачеві точні результати.

Системи, що рекомендують, відкривають нові можливості отримання персоніфікованої інформації в Інтернеті. Це також допомагає полегшити проблему перевантаження інформації, що є дуже поширеним явищем у системах пошуку інформації, і дозволяє користувачам мати доступ до послуг, які не є доступними для користувачів у інших системах. У цій роботі було обговорено дві традиційні методики рекомендацій та висвітлено їх сильні сторони та проблеми за допомогою різноманітних видів стратегій гібридизації, що використовуються для покращення їх ефективності. Обговорювались різні алгоритми навчання, що використовуються при формуванні рекомендаційних моделей та метрик оцінки, що використовуються для вимірювання якості та ефективності рекомендаційних алгоритмів. Ці знання розширяють можливості дослідників і слугуватимуть дорожньою картою для вдосконалення сучасних технічних рекомендацій. Після аналізу існуючих методів

створення рекомендаційних систем та аналогів веб-сайтів, що використовують рекомендаційні системи, була створена рекомендаційна система за допомогою методів машинного навчання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рекомендаційна система. веб-сайт. URL: <https://www.wiki.ukua.nina.az/> (дата звернення: 10.05.2022)
2. Спільна фільтрація. веб-сайт. URL: [https://hmn.wiki/uk/Collaborative\\_filtering](https://hmn.wiki/uk/Collaborative_filtering) (дата звернення 10.05.2022)
3. Netflix prize rules. веб-сайт. URL: <https://www.snellman.net/blog/archive/2006-10-15-netflix-prize.html> (дата звернення: 20.04.2022)
4. Колаборативна фільтрація веб-сайт. URL: <https://uk.wikipedia.org/wiki/>. (дата звернення: 10.05.2022)
5. Чому так складно вибрати, яке кіно подивитися (і нейромережі цю проблему не вирішать) веб-сайт. URL: <https://habr.com/ru/post/462615/> (дата звернення: 10.05.2022)
6. ЯК ПЕРСОНАЛЬНІ РЕКОМЕНДАЦІЇ ДОЗВОЛЯЮТЬ ЗБІЛЬШИТИ КОНВЕРСІЮ САЙТУ. веб-сайт. URL: <https://marketer.ua/ua/personalized-product-recommendations-allow-you-to-increase-conversion/#>.(дата звернення: 20.04.2022)
7. Глибовец, Н.Н. Создание рекомендационной системы учебного типа с использованием фреймворка / Н.Н. Глибовец, М.О. Сидоренко // Проблемы интеллектуализации компьютера : сб. ст. / Институт кибернетики им. В.М. Глушкова НАН Украины. – Киев, 2012. С. 176 – 181.
8. Джонс, М. Рекомендаційні системи: Частина 1. Введення в підходи і алгоритми. веб-сайт. URL: <https://developer.ibm.com/> (дата звернення: 01.05.2022).
9. Що таке Java. веб-сайт. URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення: 01.05.2022)
10. Що таке технологія Java та її застосування? веб-сайт. URL: [https://www.java.com/ru/download/help/whatis\\_java.html](https://www.java.com/ru/download/help/whatis_java.html) (дата звернення: 01.05.2022)

11. Основи програмування мовою Java. веб-сайт. URL: <https://ami.lnu.edu.ua/wp-content/uploads/2017/05/Java.pdf> (дата звернення: 01.05.2022)
12. Java та AI. Чи можна написати штучний інтелект Java? веб-сайт. URL: <https://javarush.ru/groups/posts/2558-java-i-ai-mozhno-li-napisatjh-iskusstvennihy-intellekt-na-java> (дата звернення: 03.05.2022)
13. Montaner, M., López, B., de la Rosa, J.L.: A taxonomy of recommender agents on the internet. Artificial Intelligence Review 19(4), С. 285–330 (2003).
14. Artificial Intelligence in Retail – 10 Present and Future Use Cases. веб-сайт. URL: <https://emerj.com/ai-sector-overviews/artificial-intelligence-retail/> (дата звернення: 11.05.2022).
15. Neuroph. веб-сайт. URL: <https://uk.m.wikipedia.org/wiki/Neuroph> (дата звернення: 11.05.2022).
16. Якою мовою пишуть штучний інтелект. веб-сайт. URL: <https://alternativescience.net/programming/137-yakoyu-movoyu-pyshut-shtuchnyj-intelekt/> (дата звернення: 11.05.2022).
17. Свінг (Java) - Swing (Java)Енциклопедія. веб-сайт. URL: [https://wikiukuk.top/wiki/Swing\\_\(Java\)](https://wikiukuk.top/wiki/Swing_(Java)) (дата звернення: 11.05.2022).
18. Apache Jena Википедія. веб-сайт. URL: [https://wiki5.ru/wiki/Apache\\_Jena](https://wiki5.ru/wiki/Apache_Jena) (дата звернення: 12.05.2022).
19. PowerLoom Knowledge Representation & Reasoning System. веб-сайт. URL: <https://www.isi.edu/isd/LOOM/PowerLoom/> (дата звернення: 12.05.2022).
20. D3web. веб-сайт. URL: <https://ru.wikibrief.org/wiki/D3web> (дата звернення: 12.05.2022).
21. Java-ML. веб-сайт. URL: <http://java-ml.sourceforge.net/>(дата звернення: 13.05.2022).
22. Weka. веб-сайт. URL: <https://uk.m.wikipedia.org/wiki/Weka> (дата звернення: 13.05.2022).

23. RapidMiner. веб-сайт. URL: <https://soware.ru/products/rapidminer> (дата звернення: 13.05.2022).
24. Jenetics веб-сайт. URL: <https://jenetics.io/>(дата звернення: 13.05.2022).
25. Watchmaker Framework. веб-сайт. URL: <https://watchmaker.uncommons.org/> (дата звернення: 14.05.2022).
26. Covington, P., J. Covington and E. Sargin, 2016. Deep Neural Networks for YouTube Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems, ACM, pp: 191–198.
27. Законодавство України про охорону праці // Збірник нормативних документів у 4 т. -К.: Держнагляд охорони праці; Основа, 2006 р (дата звернення: 26.05.2022).
28. Опалення, вентиляція та кондиціонування: накази Міністерства регіонального розвитку, будівництва та житлово комунального господарства України від 01 січ. 2014 р. № 410. веб-сайт. URL: <http://kbu.org.ua/assets/app/documents/dbn2/100.1>. (дата звернення: 26.05.2022).
29. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. веб-сайт. URL: <http://zakon3.rada.gov.ua/laws/show/z0508-18> (дата звернення: 27.05.2022).
30. Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин: Закон України від 14 лют. 2018 р. № 207. веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/z0293-10#Text> (дата звернення: 26.05.2022).
31. Оцінка категорії важкості праці. StudFiles: вебсайт. URL: <https://studfile.net/> (дата звернення: 26.05.2022).
32. ДСТУ 2293-99 Охорона праці. Терміни та визначення основних понять. Київ - 1999 р.
33. Москальова В. М. Основи охорони праці. Підручник. - Київ: ВД Професіонал, 2005.-666 с.



34. Гандзюк М. П., Желібо Е. П., Халімовський М. О. Основи охорони праці / За ред.. Гандзюка М. П. - К.: Каравела 2003 - 405 с.
35. Ткачук К. Н., Халімовський М. О., Зацарний В.В., та інші. Основи охорони праці: Підручник. -К.: Основа, 2006. -444 с.
36. Жидецький В.Ц. Основи охорони праці: Підручник. - К.: Основа, 2002. - 320 с.
37. Державні санітарні правила і норми роботи з ВДТ ЕОМ ДСанПІН 3.3.2.007-98. вебсайт. URL: <http://mozdocs.kiev.ua/view.php?id=2445> – Загол. з екрану. (дата звернення: 28.05.2022).
38. СНиП II-4-79. Природне і штучне освітлення. вебсайт. URL: [https://dnaop.com/html/45036/doc-СНиП\\_II-4-79](https://dnaop.com/html/45036/doc-СНиП_II-4-79) - Загол. з екрану. (дата звернення: 28.05.2022).
39. The Movies Dataset // Kaggle вебсайт. URL: <https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset> (дата звернення: 08.05.2022).
40. Машинне навчання для Java-розробників. вебсайт. URL: <https://javarush.ru/groups/posts/469-mashinnoe-obuchenie-dlja-java-razrabotchikov-ch1>(дата звернення: 08.05.2022).
41. Рисунок 1 - Collaborative Filtering and Content - based Filtering. вебсайт. URL:<https://data-flair.training/blogs/data-science-r-movie-recommendation/amp/>(дата звернення: 08.05.2022).
42. Рисунок 2 – Наглядний приклад рекомендації на основі спільної фільтрації. вебсайт. URL:<https://data-flair.training/blogs/data-science-r-movie-recommendation/amp/>(дата звернення: 08.05.2022).
43. Колаборативна фільтрація. вебсайт. URL: <https://uk.wikipedia.org/wiki> (дата звернення: 08.05.2022).

## ДОДАТОК А

### Код програмної реалізації

#### *AllFilters.java*

```
import java.util.ArrayList;

public class AllFilters implements Filter {
    ArrayList<Filter> filters;

    public AllFilters() {
        filters = new ArrayList<Filter>();
    }

    public void addFilter(Filter f) {
        filters.add(f);
    }

    @Override
    public boolean satisfies(String id) {
        for(Filter f : filters) {
            if (! f.satisfies(id)) {
                return false;
            }
        }

        return true;
    }
}
```

#### *DirectorsFilter.java*

```
import java.util.*;

public class DirectorsFilter implements Filter{
    private String myDirectors;

    public DirectorsFilter(String directors){
        myDirectors = directors;
    }
}
```

@Override

```
public boolean satisfies(String id){
    // String curDirectors = MovieDatabase.getDirector(id);
    // return myDirectors.contains(curDirectors);
    ArrayList<String> dirList = new ArrayList(Arrays.asList(myDirectors.split(",")));
    for(String dir:dirList){
        if(MovieDatabase.getDirector(id).contains(dir)){
            return true;
        }
    }
    return false;
}
}
```

### ***EfficientRater.java***

```
import java.util.*;

public class EfficientRater implements Rater {
    private String myID;
    //private ArrayList<Rating> myRatings;
    private HashMap<String,Rating> myRatings;

    public EfficientRater(String id) {
        myID = id;
        //myRatings = new ArrayList<Rating>();
        myRatings = new HashMap<String,Rating>();
    }
    public void addRating(String item, double rating) {
        //myRatings.add(new Rating(item,rating));
        myRatings.put(item,(new Rating(item,rating)));
    }
    public boolean hasRating(String item) {
        /* old method
        for(int k=0; k < myRatings.size(); k++){
            if (myRatings.get(k).getItem().equals(item)){
```

```
        return true;
    }
}

return false;
*/
return myRatings.containsKey(item);
}

public String getID() {
    return myID;
}

public double getRating(String item) {
    /*
    for(int k=0; k < myRatings.size(); k++){
        if (myRatings.get(k).getItem().equals(item)){
            return myRatings.get(k).getValue();
        }
    }

    return -1; */

    if(hasRating(item)){
        return myRatings.get(item).getValue();
    }
    else{
        return -1;
    }
}

public int numRatings() {
    return myRatings.size();
}

public ArrayList<String> getItemsRated() {
    ArrayList<String> list = new ArrayList<String>();

    /*
    for(int k=0; k < myRatings.size(); k++){
        list.add(myRatings.get(k).getItem());
    }
    */
}
```

```

*/
for(String movieID:myRatings.keySet()){
    list.add(movieID);
}
return list;
}
}

```

### ***JavaMLTest.java***

```

import net.sf.javaml.classification.Classifier;
import net.sf.javaml.classification.KNearestNeighbors;

import net.sf.javaml.classification.evaluation.CrossValidation;
import net.sf.javaml.classification.evaluation.PerformanceMeasure;
import net.sf.javaml.clustering.Clusterer;
import net.sf.javaml.clustering.KMeans;
import net.sf.javaml.core.*;
import net.sf.javaml.tools.InstanceTools;
import net.sf.javaml.tools.data.FileHandler;
import java.io.File;
import java.util.Map;
import java.util.Random;

public class JavaMLTest {
    public static void main(String[] args) throws Exception {
        double[] values = new double[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
        Instance instance = new DenseInstance(values,"positive");
        Instance newInstance = new SparseInstance(10);
        instance.put(1, 1.0);
        instance.put(3, 2.0);
        instance.put(7, 4.0);

        Dataset data = new DefaultDataset();
        for (int i = 0; i < 10; i++) {
            Instance tmpInstance = InstanceTools.randomInstance(25);
            data.add(tmpInstance);
        }

        data = FileHandler.loadDataset(new File("data/ratings.csv"), 0, ",");
        FileHandler.exportDataset(data,new File("data/export.csv"));
    }
}

```

```

Clusterer km = new KMeans();
Dataset[] clusters = km.cluster(data);

Classifier knn = new KNearestNeighbors(5);
knn.buildClassifier(data);

int correct = 0, wrong = 0;
for (Instance inst : data) {
    Object predictedClassValue = knn.classify(inst);
    Object realClassValue = inst.classValue();

    if (predictedClassValue.equals(realClassValue))
        correct++;
    else
        wrong++;
}

```

```

CrossValidation cv = new CrossValidation(knn);
Map<Object, PerformanceMeasure> p = cv.crossValidation(data, 5, new
Random(1));
Map<Object, PerformanceMeasure> q = cv.crossValidation(data, 5, new
Random(1));
Map<Object, PerformanceMeasure> r = cv.crossValidation(data, 5, new
Random(25));
}
}

```

### ***RecommendationRunner.java***

```

import java.util.ArrayList;

public class RecommendationRunner implements Recommender {

    public ArrayList<String> getItemsToRate(){
        ArrayList<String> moviesToRate = new ArrayList<String>();
        String[] movies = {
            "1126590", "1179031", "1198156", "1216491",
            "1361318", "1408253", "1430612", "1441953",
            "1490017", "1528100", "1571249", "1600196"
        };
        for(int i=0;i<12;i++){

```

```

    moviesToRate.add(movies[i]);
  }
  return moviesToRate;
}
public void printRecommendationsFor(String webRaterID){
  FourthRatings fr = new FourthRatings();
  int numSimilarRaters = 5;
  int minimalRaters = 3 ;
  ArrayList<Rating> recList =
fr.getSimilarRatings(webRaterID,numSimilarRaters,minimalRaters);

  if(recList.size()==0){
    printError();
  }

  else{
    printUpperPart();
    int i=0;
    for(Rating r: recList){
      i++;
      if((i+1)%2 == 0){
        System.out.println("<tr class=\"even_rows\"><td>" + i + "</td>");
      }
      else{
        System.out.println("<tr class=\"odd_rows\"><td>" + i + "</td>");
      }

      String URL = MovieDatabase.getPoster(r.getItem());
      String title = MovieDatabase.getTitle(r.getItem());
      String director = MovieDatabase.getDirector(r.getItem());
      String country = MovieDatabase.getCountry(r.getItem());
      int year = MovieDatabase.getYear(r.getItem());
      String genre = MovieDatabase.getGenres(r.getItem());
      int minutes = MovieDatabase.getMinutes(r.getItem());

      System.out.println("<td><table><tr><td class = \"pic\">");

      if(URL.length()>3){
        System.out.println("<img src = \"\"+URL+\"\" target=_blank></td>");
      }

      System.out.println("<td><h3>" + title + "</h3>");
      System.out.println("<b>by " + genre + "</b><br>");

```

```
System.out.println(year+"<br>");
System.out.println(country+"<br>");

System.out.println(minutes+" minutes</td></tr></table></td></tr>");
if(i>12) break;
}
printLowerPart();

}
}

private void printError(){
    System.out.println("This is system error, please try again!");
}
```