

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
РЕКОМЕНДУВАННЯ ФІЛЬМІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS
FOR RECOMMENDING MOVIES

Виконав: студент(ка) 2 курсу, групи КНТ-129м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Радєв П.А.

(прізвище та ініціали)

Керівник Льовкін В.М.

(прізвище та ініціали)

Рецензент Зеленьова І.Я.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Радєва Павла Анатолійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів
рекомендування фільмів
Research and Software Implementation of Methods for Recommending Movies

керівник проєкту (роботи) Льовкін Валерій Миколайович, к.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “3” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) грудень 2020 року

3. Вихідні дані до проєкту (роботи) технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Реферат, Перелік скорочень та умовних познач, Вступ, 1. Аналіз проблеми побудови систем рекомендацій, 2. Матеріали і методи, 3. Проектування програмного забезпечення, 4. Основні рішення щодо реалізації компонентів системи, 5. Експлуатація та тестування програми, 6 Економіко-організаційна частина, 7 Охорона праці та безпека в надзвичайних ситуаціях, Висновки, Перелік джерел посилання, Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Льовкін В.М., доцент		
6 Економіко-організаційна частина	Пожуєва Т.О., професор		
7 Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз та дослідження предметної області	1-2 тиждень	
2	Розробка та затвердження технічного завдання	3 тиждень	Технічне завдання
3	Дослідження методів розв’язання проблеми	4 тиждень	
4	Проектування структури програмного забезпечення	5 тиждень	Структурна схема
5	Розроблення схеми функціонування програмного забезпечення та модулів	6-7 тиждень	Функціональна схема. Модулі
6	Розроблення програмного коду	8 тиждень	Текст програми
7	Тестування та відлагодження програмного забезпечення	9 тиждень	Програма
8	Розробка розділів пояснювальної записки	10-11 тиждень	Пояснювальна записка
9	Розробка графічного матеріалу	12 тиждень	Графічний матеріал
10	Захист магістерської роботи	13 тиждень	

Студент(ка)

(підпис)

Радєв П.А.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Льовкін В.М.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
133 с., 38 рис., 7 табл., 3 дод., 29 джерел.

СИСТЕМИ РЕКОМЕНДУВАННЯ, РЕКОМЕНДУВАННЯ
КІНОФІЛЬМІВ, ДОВГА КОРОТКОЧАСНА ПАМ'ЯТЬ, WORD2VEC, C#,
PYTHON, MYSQL.

Об'єкт дослідження – процес рекомендування фільмів. Предмет дослідження – методи рекомендування фільмів.

Мета роботи – розроблення програмного забезпечення рекомендування кінофільмів для забезпечення персонального пошуку нових фільмів.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мови програмування C# і Python, система MySQL.

Результати. Виконано аналіз, проєктування та реалізацію програмної системи рекомендування фільмів, створено метод рекомендування фільмів.

Наукова новизна роботи полягає в запропонованому методі рекомендування фільмів, який характеризується виконанням рекомендування фільмів на основі використання довгої короткочасної пам'яті та використання в якості її вхідних ознак закодованих векторних представлень фільмів за допомогою використання методу word2vec, що дозволяє виконувати рекомендування фільмів з врахуванням послідовності уподобань користувача.

Висновки. Розроблено метод рекомендування фільмів та виконано програмну реалізацію системи рекомендування.

Галузь використання – програмне забезпечення перегляду фільмів, створення переліків перегляду та накопичення даних про фільми, персоналізований пошук нових фільмів.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 133 pages, 38 figures, 7 tables, 3 appendixes, 29 sources.

RECOMMENDATION SYSTEMS, MOVIE RECOMMENDATIONS, LONG SHORT-TERM MEMORY, WORD2VEC, C#, PYTHON, MYSQL.

The object of research is the process of film recommendation. The subject of research is methods of film recommendation.

The purpose of the work is to develop software to recommend films to ensure user personal search for new films.

Materials, methods and technical means: object-oriented programming, C# and Python programming languages, MySQL system.

Results. The analysis, design and implementation of the software system of film recommendation are performed, the method of film recommendation is developed.

The scientific novelty of the work lies in the proposed method of film recommendation, which is characterized by the implementation of film recommendations based on the usage of long short-term memory and usage of encoded vector representations of films using the word2vec method as input features for neural network, which allows to recommend films based on user preferences.

Conclusions. The method of film recommendation has been developed and the software implementation of the recommendation system has been performed.

Field of use is software for watching films, creating playlists and collecting data about films, personalized search for new films.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми побудови систем рекомендацій	11
1.1 Типи і принципи систем рекомендацій.....	11
1.2 Програмні системи рекомендацій фільмів	15
1.3 Висновки за розділом 1	20
2 Матеріали і методи.....	22
2.1 Аналіз методів рекомендування фільмів.....	22
2.2 Рекурентні нейронні мережі	26
2.3 Метод word2vec	29
2.4 Запропонований метод рекомендування фільмів	31
2.5 Висновки за розділом 2	34
3 Проєктування програмного забезпечення	35
3.1 Вибір засобів реалізації програмного забезпечення.....	35
3.2 Функціональні вимоги до програмного забезпечення	38
3.3 Проєктування бази даних	39
3.4 Структура програмного забезпечення	44
3.5 Висновки за розділом 3	45
4 Основні рішення щодо реалізації компонентів системи.....	46
4.1 Алгоритм функціонування системи.....	46
4.2 Опис модулів реалізованої системи	48
4.3 Висновки за розділом 4	53
5 Експлуатація та тестування програми	54
5.1 Призначення програми	54
5.2 Умови виконання програми	54
5.3 Виконання програми.....	54
6 Економіко-організаційна частина.....	65
6.1 Планування розробки програмного продукту.....	65

6.2 Визначення витрат на розробку програми	68
6.2.1 Розрахунок основної заробітної плати	69
6.2.2 Розрахунок додаткової заробітної плати	69
6.2.3 Відрахування на єдиний соціальний внесок	70
6.2.4 Визначення витрат на матеріали	70
6.2.5 Витрати на спеціальне обладнання	71
6.2.6 Інші прямі витрати	74
6.2.7 Розрахунок накладних витрат.....	75
6.3 Оцінка економічної ефективності	76
7 Охорона праці та безпека в надзвичайних ситуаціях.....	79
7.1 Аналіз потенційних небезпек.....	79
7.2 Заходи по забезпеченню безпеки	80
7.3 Заходи по забезпеченню промислової санітарії та гігієни праці	84
7.4 Заходи по забезпеченню пожежної безпеки.....	90
7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях	92
Висновки	96
Перелік джерел посилання	97
Додаток А Технічне завдання	100
Додаток Б Текст програми	105
Додаток В Слайди презентації.....	126

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ВДТ	– візуальний дисплейний термінал;
ДКП	– довга короткочасна пам'ять;
ЕОМ	– електронно-обчислювальна машина;
ПК	– персональний комп'ютер;
ПРУ	– протирадіаційне укриття;
ПУЕ	– правила улаштування електроустановок;
РНМ	– рекурентна нейронна мережа;
СКБД	– система керування базами даних;
ШНМ	– штучна нейронна мережа.

ВСТУП

Щодня мільйони людей зайняті пошуком в інтернеті: хтось шукає фільми або одяг, хтось – машину або туристичну путівку, – і всі користувачі об'єднані однією метою – знайти те, що потрібно: якщо в минулому столітті про появу нових товарів дізнавалися з поштових розсилок, потім цей процес прискорився в десятки (а то і сотні) разів за рахунок появи спочатку телебачення, а потім і інтернету, в останні рік-два одним з провідних трендів в удосконаленні найрізноманітніших пошукових систем стає застосування алгоритмів машинного навчання, як доповнення до процесу самостійного пошуку (серед мільйонів найменувань всіляких товарів і послуг), на ресурсах стали з'являтися рекомендаційні системи, які пророкують, що саме було б цікаво даному користувачеві, такі рекомендаційні алгоритми в процесі своєї роботи постійно донавчаються, адаптуються і трансформуються, з часом все краще розуміючи користувача, і в результаті свого функціонування 50% і більше рекомендованих товарів або послуг в тій чи іншій мірі задовольняють пошуковим запитам користувачів [1].

Завдання рекомендаційної системи – проінформувати користувача про продукт, який йому може бути найбільш цікавий в даний момент часу:

- клієнт отримує інформацію, а сервіс заробляє на наданні якісних послуг;
- послуги – це не обов'язково прямі продажі пропонованого товару;
- сервіс також може заробляти на комісійних або просто збільшувати лояльність користувачів, яка потім виливається в рекламні та інші доходи;
- залежно від моделі бізнесу рекомендації можуть бути його основою, як, наприклад, у TripAdvisor, а можуть бути просто зручним додатковим сервісом (як, наприклад, в якомусь інтернет-магазині), покликаним поліпшити досвід користувачів і зробити навігацію по каталогу більш зручною [2].

Грунтуючись на описаній актуальності роботи, було визначено мету роботи як розроблення програмного забезпечення рекомендаційного пошуку нових фільмів. Така персональна система пошуку дозволяє підбирати актуальні дані для користувача, що може дозволити в застосунку заробляти на рекламі, на використанні посилань для переходу на зовнішні сайти з фільмами.

1 АНАЛІЗ ПРОБЛЕМИ ПОБУДОВИ СИСТЕМ РЕКОМЕНДАЦІЙ

1.1 Типи і принципи систем рекомендацій

Рекомендаційні системи – це великий клас моделей, мета яких – підвищити бізнес-показники за рахунок релевантних рекомендацій користувачеві в правильному місці, в правильний час і через правильний канал комунікації [1].

Щодо розповсюдження рекомендаційних систем можна виділити наступне:

– існує поширений стереотип, який до цих пір заважає широкому застосуванню в бізнесі рекомендаційних систем: багатьом здається, що в реальності впровадження рекомендаційних алгоритмів – це занадто складно і вимагає глобальної перебудови всього процесу збору і обробки даних, а також зміни бізнес-процесів, логістики і так далі, багато користувачів сумнівається і не може оцінити, який же коефіцієнт повернення від інвестицій в подібній трансформації, ці сумніви абсолютно необґрунтовані, адже насправді рекомендаційні системи можуть бути корисні практично кожному бізнесу, а щоб почати рекомендувати, часто цілком достатньо тих даних, які вже збираються;

– релевантні рекомендації скорочують час, необхідний для пошуку товарів і послуг, і значно збільшують вірогідність попадання в поле зору користувача інших об'єктів, які зможуть його зацікавити: в результаті підвищується лояльність і задоволеність користувачів вебсервісами, як правило, користувачі також взаємодіють з більшою кількістю товарів, і це призводить до збільшення споживання і зростання прибутку, крім того, інформаційні бюлетені, персоналізовані рекламні матеріали і push-повідомлення спонукають користувачів повертатися, збільшують частоту відвідувань постійними користувачами і зменшують відтік клієнтів;

– сьогодні загалом кожній компанії просто необхідно налагодити процес збору даних і вміти грамотно і ефективно використовувати їх в

бізнесі, тим самим оптимізуючи і покращуючи призначений для користувача контент, знижуючи витрати, збільшуючи виручку і середній чек, і підвищуючи рентабельність бізнесу в цілому [1].

Серед всіх сучасних систем рекомендацій можна виділити наступні типи:

- статистичні системи рекомендацій – це системи, які ґрунтуються на статистичних даних, зібраних з користувачів: простіше кажучи, це звичні рейтинги найкращого (найбільш завантажувані, найпопулярніші тощо), важливою відмінністю є те, що такий підхід не є персоналізованим і пропонує однакові рекомендації для всіх користувачів;

- демографічні системи рекомендацій порівнюють характеристики об'єкта з характеристиками користувача: розглядатися можуть прості дискретні властивості (вік, стать, місце проживання, національність), а також і більш складні (наприклад, інтерес користувача до певного об'єкту), деякі з них система може легко визначити самостійно: країну і навіть місто, в якому проживає користувач, можна визначити за ір-адресою, потрібну мову можна дізнатися з властивостей браузера, використовуючи JavaScript, інші дані можна попросити користувача вказати явно – не дивлячись на те, що обсяг отриманої інформації про користувача буде досить обмежений, вона може допомогти прийняти важливі рішення;

- асоціативні системи рекомендацій будують рекомендації на основі даних про те, які об'єкти використовуються разом: яскравий приклад застосування таких систем – це аналіз покупок користувачів, наприклад, хтось купив собі новий телефон і, цілком ймовірно, йому знадобляться до нього навушники, чохол, зарядка, додаткова карта пам'яті і інші аксесуари, інші системи рекомендацій тут можуть бути безсилі, адже у цих об'єктів немає ніяких загальних параметрів і порівняти їх не представляється можливим, їх об'єднує лише те, що вони використовуються разом;

- інформаційні системи рекомендацій – це системи, які шукають об'єкти, які схожі на ті, які користувач уже позитивно оцінив: на відміну від

асоціативних систем вони можуть оперувати практично будь-якими даними (від найпростіший значень (купив – не купив) до складних текстових описів);

– колективні (колаборативні) системи рекомендацій – найпоширеніші системи, які для передбачення рейтингу певного об'єкта керуються оцінками інших користувачів: хоча такі системи досить ефективні, їх точність сильно залежить від того, на скільки перетинаються оцінювані об'єкти між окремими користувачами, чим більші і частіші ці перетини, тим точніше буде система, цей факт обмежує сферу застосування даної моделі (якщо кожен користувач буде мати свою унікальну інформацію, яку ніхто інший не бачив, колективні системи рекомендацій будуть безсилі) [3].

Окрім цих типів існує ще один – гібридний.

У гібридних рекомендаційних системах зустрічаються такі типи комбінування:

– реалізація окремо колаборативного і тематичних алгоритмів і об'єднання їх припущень;

– включення деяких тематичних правил в колаборативну методику;

– включення деяких колаборативних правил в контентну методику;

– побудова загальної моделі, що включає в себе правила обох методик

[1].

Дві основні методики збору даних:

– явний збір даних: в цьому випадку користувач добровільно надає системі необхідні для роботи дані, явний збір даних може полягати в пропозиції користувачу поставити диференційовану оцінку того чи іншого товару, створити список «улюблених продуктів» або відповісти на запитання профілю, основний недолік алгоритму стане очевидний, якщо загадати, наскільки дратує часом нав'язливе прохання поставити оцінку тому чи іншому мобільному застосунку, а вже якщо користувачі так неохоче діляться такою інформацією, то їх реакція на пропозицію заповнити профіль якимись персональними даними може тим більше бути негативною;

– неявний збір інформації – ця методика збору даних являє собою своєрідне «шпигунство» за користувачем: неявний збір інформації може відбуватися за допомогою спостереження за поведінкою користувача – включаючи його покупки і оцінки в інших інтернет-магазинах, а також збір інформації безпосередньо з комп'ютера клієнта, такі методики везуть за собою своєрідний «багаж» етичних проблем [4].

Основні фактори, які можна використати для порівняння систем рекомендацій:

– прихильність до користувача означає, що системі необхідно ідентифікувати користувача, щоб давати йому поради: очевидно, що цією проблемою страждають всі персоналізовані системи рекомендацій, вирішити цю проблему можна простою авторизацією або іншими методами ідентифікації користувача;

– прихильність до об'єкта позначає контекст, в якому можна застосувати певну систему рекомендацій: для деяких систем, наприклад, для асоціативної і інформаційної потрібно, щоб користувач вибрав конкретний об'єкт, від якого вона зможе відштовхуватися (шукати інші об'єкти, що схожі або доповнюють обраний), інші ж системи можуть дати рекомендації в будь-якому контексті на будь-яку тему;

– рішення проблеми нового об'єкта – це одна з варіацій проблеми холодного початку: використовувати тільки що доданий новий об'єкт можуть тільки системи, що використовують безпосередньо його властивості, які, зазвичай, задаються при створенні, для статистичних та колективних систем рекомендацій буде потрібний якийсь час, поки його не оцінить достатня кількість користувачів, для асоціативних систем ця проблема може бути складніше, так як доведеться шукати нові шаблони використання об'єкта;

– рішення проблеми нового користувача – це інша варіація проблеми холодного початку, тільки в даному випадку нічого не відомо про користувача: очевидно, що це не буде проблемою для асоціативних і статистичних систем рекомендацій, так як вони взагалі не залежать від

користувача: для демографічної системи це не буде проблемою в тому випадку, якщо при створенні користувача були вказані хоч якісь дані, які система зможе використовувати, для колективної ж системи знадобиться час, поки вона зможе дізнатися потрібну їй інформацію;

– рішення проблеми перетину об'єктів – характерно практично для всіх систем рекомендацій, крім колективної і частково асоціативної: їх ефективність залежить від того як часто перетинаються порівнювані об'єкти, для колективної системи це означає, що чим більше людей перегляне якийсь матеріал, тим точнішу думку вона зможе про нього скласти, для асоціативної – що чим частіше об'єкт буде перетинатися з якимось вузьким набором інших об'єктів, тим простіше буде системі виявити шаблони його застосування, якщо ж об'єкт буде перетинатися з дуже великою кількістю об'єктів, то це тільки заплутає систему;

– врахування якісних аспектів – це пункт, який відповідає за те, чи враховує система якість об'єкта: більшість методів, які шукають схожість між об'єктами, не враховують їх якість, вони можуть знайти дві схожі новини, але якщо одна з них цікава, то це ще не означає, що і друга теж, у такому випадку система не може бути впевнена, що те, що вона рекомендує користувачеві, має цінність, цю проблему можуть вирішити тільки ті системи, які враховують рейтинги об'єктів, тобто колективні та статистичні, при чому рейтинги можуть бути отримані як і явними методами так і не явними спостереженнями [3].

1.2 Програмні системи рекомендацій фільмів

Системи рекомендацій фільмів найчастіше побудовані навколо сховища даних про фільми або всередині системи перегляду фільмів.

Однією з таких систем є IMDb [5]. IMDb надає користувачам всеосяжну базу даних про фільми, телесеріали, акторів і актрис, режисерів, продюсерів і про все, що пов'язано зі світом кіно, це найбільша в світі база

даних про кінематограф, належить компанії Amazon: можна дивитися фотографії, тизери та трейлери, читати огляди та відгуки інших користувачів, переглядати топ-рейтинги і номінації, шукати нестиковки і кіно-ляпи, шукати найближчий до себе кінотеатр і переглядати доступні сеанси, читати останні новини про кіно з сотень засобів масової інформації, дізнаватися, що вийде на екрани в найближчому майбутньому і новітні релізи на DVD, читати відгуки користувачів і докладні огляди кінокритиків, надається історія пошуку і переглядів з сайту IMDb [6].

Інтерфейс вебзастосунку imdb показано на рис. 1.1.

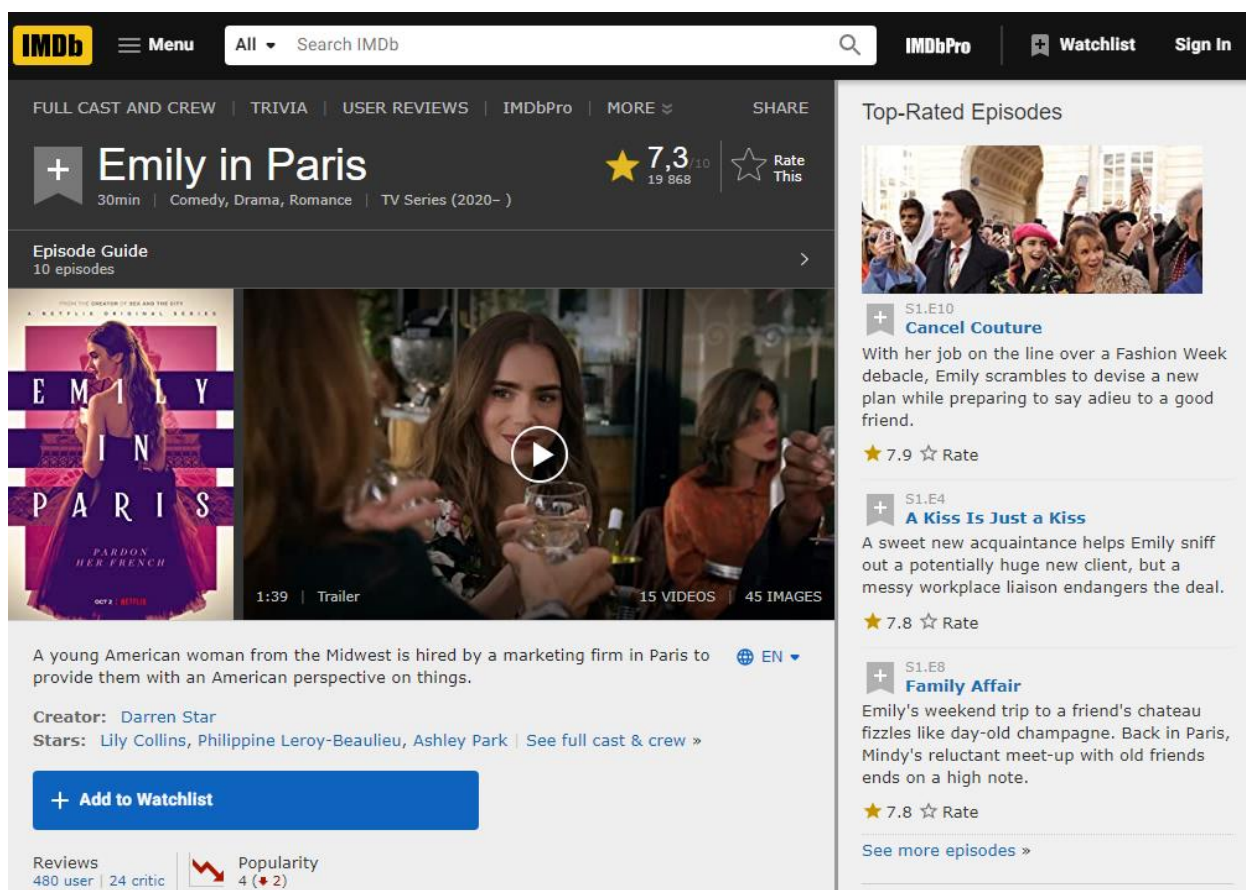


Рисунок 1.1 – Інтерфейс вебсервісу imdb

Вебзастосунок дозволяє виконувати наступні функції:

- переглядати інформацію про наявні фільми;
- створювати власний перелік фільмів для перегляду;
- виставляти оцінку фільму;

- переглядати переліки фільмів (найкращі за рейтингом, найбільш популярні, за жанрами тощо);
- створювати власні рецензії та переглядати існуючі;
- переглядати перелік власних рейтингів, рецензій;
- переглядати перелік подібних фільмів на обраний.

Приклад роботи системи рекомендацій наведено на рис. 1.2.

More Like This

[Learn more](#)

Ретчед (TV Series 2020)

Crime | Drama | Mystery

★★★★★ 7,3/10

In 1947, Mildred Ratched begins working as a nurse at a leading psychiatric hospital. But beneath her stylish exterior lurks a growing darkness.

[Add to Watchlist](#) [Next »](#)

Stars: Sarah Paulson, Finn Wittrock,...

Storyline

[Edit](#)

A young American woman from the Midwest is hired by a marketing firm in Paris to provide them with an American perspective on things.

[Plot Summary](#) | [Add Synopsis](#)

Plot Keywords: [marketing](#) | [american in paris](#) | [character name in series title](#) | [forename in series title](#) | [american](#) | [See All \(9\)](#) »

Taglines: Pardon her French.

Genres: [Comedy](#) | [Drama](#) | [Romance](#)

Certificate: [See all certifications](#) »

Parents Guide: [View content advisory](#) »

Рисунок 1.2 – Система рекомендацій imdb

Загалом застосунок надає стільки інформаційних можливостей, що відносно створення рекомендацій він є достатньо перевантаженим, адже головним його акцентом є база фільмів, а функції щодо рекомендацій для майбутнього перегляду дещо губляться у всьому масиві наявних інформаційних можливостей. Великою перевагою imdb є те, що база фільмів містить майже кожен існуючий в світі фільм, і в цих можливостях будь-якому застосунку конкурувати з imdb дуже важко.

Українським аналогом всесвітньо відомого сервісу є Moviestape [7]. Інтерфейс застосунку продемонстровано на рис. 1.3.

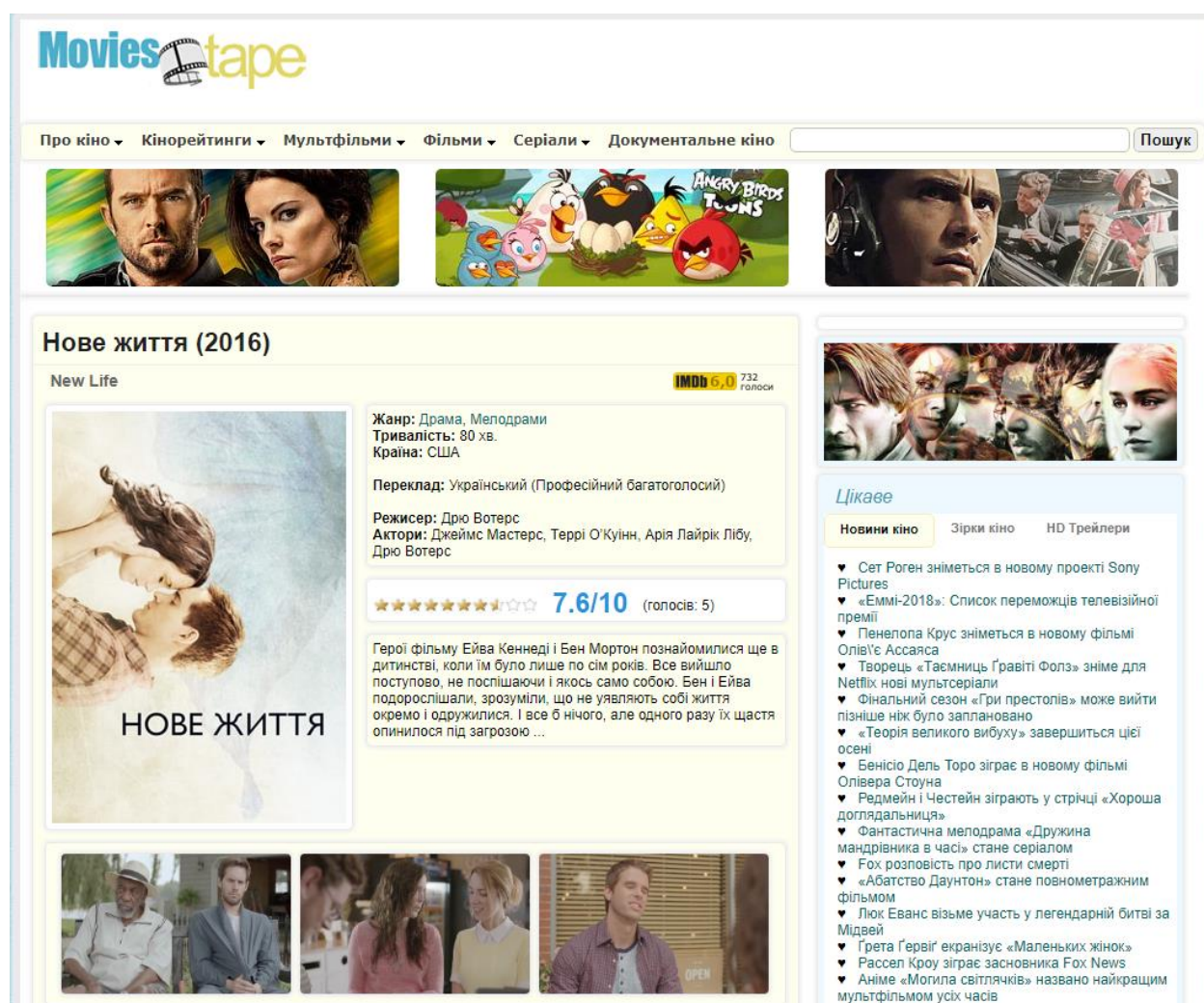


Рисунок 1.3 – Інтерфейс вебсервісу Moviestape

Moviestape надає можливість виконувати наступні функції:

– переглядати трейлери фільмів;

- переглядати різні групи фільмів;
- отримувати перелік подібних фільмів (рис. 1.4);
- переглядати інформацію про фільм;
- оцінювати фільм;
- створювати коментар до фільму.



Рисунок 1.4 – Система рекомендацій Moviestape

Сервіс Netflix [8] окрім отримання інформації про фільми дозволяє переглядати самі фільми, що загалом як раз є його основною функцією, при цьому перегляд вимагає платної підписки.

Застосунок Netflix (рис. 1.5) дозволяє в розділ «Мій список» додавати будь-які відео, щоб потім продовжити перегляд, а також для того, щоб Netflix пропонував більше подібного контенту, при цьому будь-якому фільму можна поставити вподобання або невподобу, будь-яке відео можна завантажувати для подальшого перегляду без доступу до Інтернет (для цього достатньо натиснути відповідну іконку праворуч від конкретного епізоду або кнопку Download біля фільму чи мультфільму), але дивитися завантажені відео можна тільки в додатку Netflix і тільки з активною підпискою [9].



Рисунок 1.5 – Інтерфейс Windows-застосунку Netflix [10]

1.3 Висновки за розділом 1

Розглянуто основні та гібридні типи рекомендаційних систем, фактори, за якими можна порівнювати системи рекомендацій, основні маркери сучасних особливостей розповсюдження систем рекомендацій, методики збору даних.

Проаналізовано сучасні програмні системи рекомендацій фільмів, які використовуються в сервісах IMDb, Netflix, Moviestape.

Основною функцією окрім отримання даних про фільм для забезпечення необхідності рекомендацій є функція створення переліку для перегляду та функція виставлення оцінки фільми, на чому і базується в результаті робота системи рекомендацій.

До завдань дипломної кваліфікаційної роботи магістра належать наступні завдання:

- дослідження методів рекомендування фільмів;
- розробка технічного завдання;
- вибір засобів розробки;

- проєктування структури програмного забезпечення;
- розробка програмного забезпечення рекомендації фільмів;
- тестування та відлагодження програми;
- економічне обґрунтування доцільності розробки.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Аналіз методів рекомендації фільмів

Використання методів інтелектуального аналізу даних загалом змінило способи, якими тепер зазвичай оперують методи рекомендації фільмів, серед усіх підходів загалом можна виділити наступні:

– асоціативні правила: алгоритми асоціативних правил витягують правила, що передбачають появу елемента на основі присутності інших елементів у транзакції, зокрема, наприклад, враховуючи набір транзакцій, де кожна транзакція є сукупністю елементів, асоціативне правило застосовує формулу $A \rightarrow B$, де A і B – це два набори елементів, асоціативні правила можуть сформувати дуже компактне представлення даних, що може підвищити ефективність зберігання, а також продуктивність, крім того відома ефективність асоціативних правил щодо керування персоналізованими маркетинговими рішеннями, також існує чіткий взаємозв'язок між цим методом та метою рекомендаційної системи, але загалом варто відзначити, що ці методи вони не стали основними;

– кластеризація: методи кластеризації застосовуються в різних сферах, таких як розпізнавання образів, обробка зображень, статистичний аналіз даних та виявлення знань, алгоритм кластеризації намагається розділити набір даних на набір підкластерів, щоб виявити значущі групи, що існують у них, після формування кластерів думки інших користувачів кластеру можуть бути усереднені та використані для надання рекомендацій окремим користувачам, хороший метод кластеризації дасть кластери високої якості, у яких подібність між елементами всередині кластерів висока, тоді як подібність між кластерами низька, у деяких підходах до кластеризації користувач може частково брати участь у різних кластерах, і тоді рекомендації базуються на середньому значенні кластерів участі, яке зважається за ступенем участі, Методи K-середніх та самоорганізуючі карти є найбільш часто використовуваними серед різних методів кластеризації

(метод К-середніх приймає вхідний параметр, а потім розділяє набір з n елементів на K кластерів, карти самоорганізації – це метод некерованого навчання, заснований на техніці кластеризації штучних нейронів), методи кластеризації можуть бути використані для зменшення набору кандидатів;

– дерева рішень: дерева рішень базуються на методології графів дерев, яка побудована шляхом аналізу набору навчальних прикладів, для яких відомі мітки класів, потім вони застосовуються для класифікації прикладів, які не використовувались під час навчання, при цьому якщо вони навчаються на дуже високоякісних даних, вони можуть робити дуже точні прогнози, дерева рішень є більш зрозумілими, ніж інші класифікатори, такі як Support Vector Machine та нейронні мережі, оскільки вони зрозуміло поєднують прості питання щодо даних, дерева рішень також гнучкі в обробці зразків із поєднанням ознак з реальними та категоріальними значеннями, а також зразків, які мають деякі специфічні відсутні ознаки;

– штучні нейронні мережі (ШНМ): ШНМ – це структура багатьох з'єднаних нейронів (вузлів), які систематично розташовані шарами, зв'язки між нейронами мають ваги, пов'язані з ними, залежно від ступеня впливу одного нейрона на інший, є деякі переваги використання нейронних мереж у деяких особливих проблемних ситуаціях, наприклад, завдяки тому, що ШНМ містить багато нейронів, а також присвоює вагу кожному з'єднанню, ШНМ є досить надійною щодо шумних та помилкових наборів даних, ШНМ має здатність оцінювати нелінійні функції та фіксувати складні взаємозв'язки в наборах даних, ШНМ також можуть бути ефективними і навіть працювати, якщо частина мережі виходить з ладу, основним недоліком є те, що важко знайти ідеальну топологію мережі для даної проблеми;

– аналіз посилань: аналіз посилань – це процес побудови мереж взаємопов'язаних об'єктів з метою вивчення закономірностей та тенденцій, це дало великий потенціал для вдосконалення вебпошуку, більшість алгоритмів аналізу посилань обробляють вебсторінку як один єдиний вузол у вебграфі;

– регресія: регресійний аналіз використовується тоді, коли вважається, що дві або більше змінних систематично пов'язані лінійною залежністю, це потужний і різноманітний процес для аналізу асоціативних зв'язків між залежною змінною та однією або кількома незалежними змінними, використання регресії виконує прогнозування та тестування систематичних гіпотез про взаємозв'язки між змінними, крива може бути корисною для виявлення тенденції в наборі даних, хоч вона є лінійною, параболічною чи якоїсь іншої форми;

– байєсівські класифікатори є ймовірнісною основою для вирішення задач класифікації, яка базується на визначенні умовної ймовірності та теоремі Байєса, байєсівські класифікатори розглядають кожен атрибут та мітку класу як випадкові величини: для запису з N ознак $(A_1, A_2, \dots, A_N)$, метою класифікатора є прогнозування класу C_k шляхом знаходження значення C_k , яке максимізує ймовірність класу з урахуванням даних $P(C_k | A_1, A_2, \dots, A_N)$ шляхом застосування теореми Байєса $P(C_k | A_1, A_2, \dots, A_N) \propto P(A_1, A_2, \dots, A_N | C_k)P(C_k)$, найбільш часто використовуваний байєсівський класифікатор відомий як наївний класифікатор Байєса: для оцінки умовної ймовірності $P(A_1, A_2, \dots, A_N | C_k)$ даний класифікатор передбачає імовірнісну незалежність атрибутів, тобто наявність або відсутність певного атрибута не пов'язана з наявністю або відсутністю будь-якого іншого, це припущення веде до того, що $P(A_1, A_2, \dots, A_N | C_k) = P(A_1 | C_k) P(A_2 | C_k) \dots P(A_N | C_k)$, основні переваги таких класифікаторів полягають у тому, що вони стійкі до ізольованих точок шуму та нерелевантних атрибутів, і вони обробляють відсутні значення, ігноруючи екземпляр під час розрахунків оцінки ймовірності, однак припущення про незалежність може не виконуватися для деяких атрибутів, оскільки вони можуть корелювати, – у цьому випадку звичайним підходом є використання байєсівських мереж, байєсівські класифікатори можуть виявитись практичними для середовищ, в яких знання про уподобання користувачів змінюються повільно щодо часу, необхідного для побудови моделі, але не підходять для середовищ, в яких моделі переваг

користувачів повинні швидко або часто оновлюватися, метод також має успіх у системах рекомендацій на основі моделей, оскільки часто використовується для вироблення моделі для систем рекомендацій на основі вмісту;

– методи заповнення матриці: суть техніки заповнення матриці полягає у передбаченні невідомих значень за допомогою матриць елементів користувача, метод К-найближчих сусідів на основі кореляції є одним з основних методів, що застосовуються в системах рекомендацій щодо спільної фільтрації, вони значною мірою залежать від історичних даних про рейтинг користувачів щодо оцінюваних елементів, здебільшого рейтингова матриця завжди дуже велика і розріджена через те, що користувачі не оцінюють більшість елементів, представлених у матриці, ця проблема завжди призводить до нездатності системи давати надійні та точні рекомендації користувачам, на практиці використовуються різні варіації моделей низького рангу для заповнення матриці, особливо щодо застосування в спільній фільтрації, найбільш широко використовуваний на практиці алгоритм для розв'язання цієї проблеми – це мінімізація методом найменших квадратів, в одній з робіт було запропоновано використовувати метод доповнення матриці в задачі Netflix як практичний приклад використання цієї методики, в іншій було запропоновано використовувати техніку Singular Value Decomposition в алгоритмі OptSpace для вирішення проблеми заповнення матриці, результат їх експерименту показав, що Singular Value Decomposition здатна забезпечити надійну початкову оцінку охоплюючого підпростору, яка може бути додатково уточнена градієнтним спуском, при цьому необхідно відзначити, що методи, засновані на моделях, вирішують проблему розрідженості, основним недоліком методів є те, що процес побудови моделі є обчислювально дорогим і ємність використання пам'яті дуже інтенсивна, крім того, вони не полегшують проблему холодного старту [11].

2.2 Рекурентні нейронні мережі

Рекурентні нейронні мережі (РНМ) – це мережі прямого поширення сигналу, але з особливістю: нейрони отримують інформацію не тільки від попереднього шару, але і від самих себе попереднього проходу (рис. 2.1), це означає, що порядок, в якому подаються дані і навчається мережа, стає важливим, при цьому великою складністю мереж РНМ є проблема зникаючого градієнта, яка полягає у швидкій втраті інформації з плином часу, звичайно, це впливає лише на ваги, а не стан нейронів, але ж саме в них накопичується інформація, зазвичай мережі такого типу використовуються для автоматичного доповнення інформації [12].

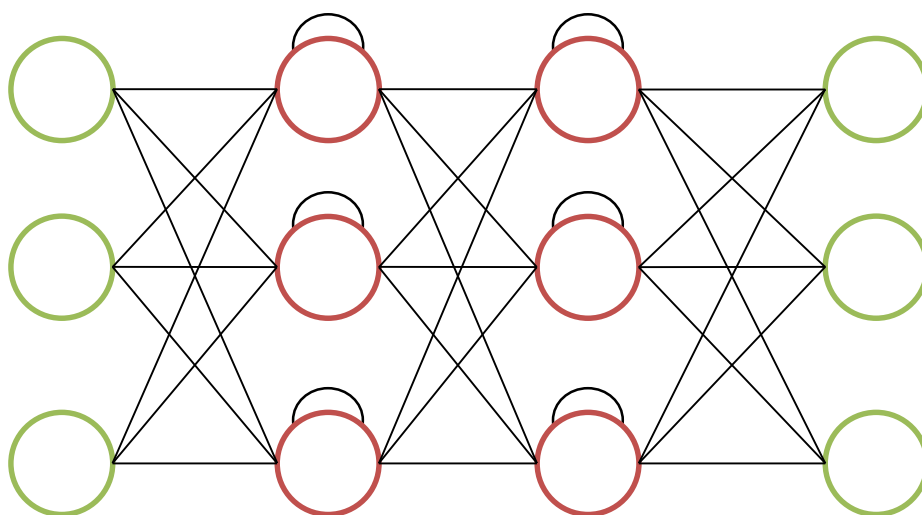


Рисунок 2.1 – Архітектура РНМ

Одним з різновидів РНМ є довга короткочасна пам'ять (ДКП).

ДКП має аналогічний потік управління, як і РНМ, при цьому відмінності полягають у операціях всередині комірок ДКП (рис. 2.2), ці операції використовуються, щоб дозволити ДКП зберігати або забувати інформацію [13].

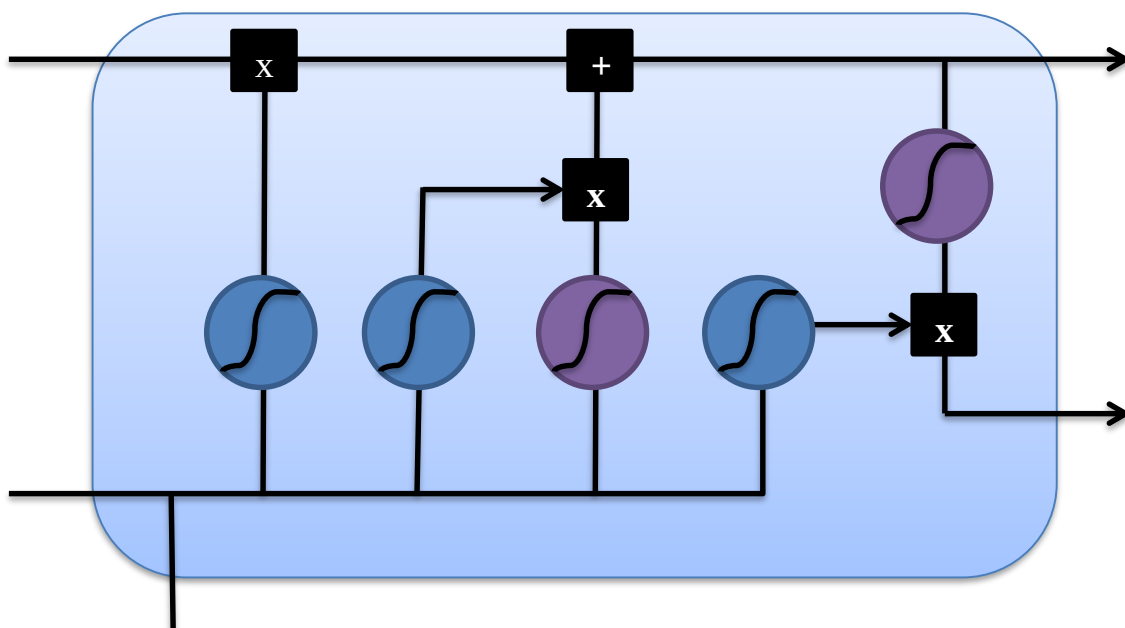


Рисунок 2.2 – Архітектура нейронної мережі з ДКП

Логіка роботи ДКП ґрунтується на наступному:

- основною концепцією ДКП є стан комірки і її різні ворота, стан комірки діє як транспортна магістраль, яка передає відносну інформацію на всьому шляху ланцюга послідовності, відповідно можна сприймати це як “пам’ять” мережі, теоретично стан комірки може нести відповідну інформацію протягом усієї обробки послідовності, тож навіть інформація з попередніх часових кроків може перейти до наступних часових кроків, зменшуючи ефект короточасної пам’яті, поки стан комірки передається, інформація додається або видаляється до стану комірки через ворота (ворота – це різні нейронні мережі, які вирішують, яка інформація дозволена про стан комірки, ворота можуть навчитись, яку інформацію необхідно зберегти чи забути);

- ворота містять сигмоїдні функції активації: сигмоїдна активація подібна до активації $\tanh$, але замість того, щоб зменшувати значення від -1 до 1, вона зменшує значення в інтервал від 0 до 1, що корисно для оновлення або забуття даних, оскільки будь-яке число, помножене на 0, дорівнює 0, змушуючи значення зникати або "забуватися", будь-яке число, помножене на

1, є однаковим значенням, тому це значення залишається незмінним або "зберігається", мережа може дізнатись таким чином, які дані не є важливими, тому їх можна забути, або які дані є важливими і які відповідно потрібно зберігати;

– ворота забуття вирішують, яку інформацію слід викинути чи зберегти: інформація з попереднього прихованого стану та інформація з поточного вводу передається через сигмоподібну функцію, в результаті отримуються значення від 0 до 1 (ближче до 0 означає забути, а ближче до 1 означає зберегти);

– вхідні ворота: для оновлення стану комірки використовуються вхідні ворота, спочатку передається попередній прихований стан і поточний вхід у сигмоподібну функцію, це вирішує, які значення будуть оновлені, перетворюючи значення в інтервал між 0 і 1 (0 означає неважливе, а 1 означає важливе), також передається прихований стан і поточний вхід у функцію $\tanh$, щоб звести значення до інтервалу від -1 до 1, щоб допомогти регулювати мережу, потім помножується вихід $\tanh$ на сигмоподібний вихід, сигмоподібний вихід вирішує, яку інформацію важливо зберегти з $\tanh$ -виводу;

– стан комірки, тепер за результатами попередніх обчислень повинно бути достатньо інформації для обчислення стану комірки: по-перше, стан комірки точково помножується на вектор забуття, це надає можливість скинути значення в стані комірки, якщо їх помножити на значення, близькі до 0, потім беруться вихідні дані з вхідних воріт і робиться точкове додавання, яке оновлює стан комірки до нових значень, які нейронні мережі вважають актуальними, це дає новий стан комірки;

– вихідні ворота: вихідні ворота вирішують, яким має бути наступний прихований стан, адже прихований стан містить інформацію про попередні входи, прихований стан також використовується для прогнозування, спочатку передається попередній прихований стан і поточний вхід у сигмоподібну функцію, потім передається нещодавно модифікований стан

комірки функції $\tanh$, множиться вихід $\tanh$ на сигмоподібний вихід, щоб вирішити, яку інформацію повинен нести прихований стан, результатом є прихований стан, потім новий стан комірки та новий прихований стан переносяться на наступний часовий крок [13].

2.3 Метод word2vec

Word2vec – це інструмент для розрахунку векторних представлень слів, на вхід якого подається корпус тексту, а на виході виходить набір векторів слів, принцип роботи:

- знаходження зв'язків між контекстами слів згідно з припущенням, що слова, що знаходяться в схожих контекстах, мають тенденцію означати схожі речі, тобто бути семантично близькими, при цьому більш формально завдання стоїть так: максимізація косинусної близькості між векторами слів (скалярний добуток векторів), які з'являються поруч один з одним, і мінімізація косинусної близькості між векторами слів, що не з'являються поруч один з одним, поруч один з одним в даному випадку означає в близьких контекстах;

- спершу читається корпус, і розраховується частота кожного слова в корпусі (тобто кількість разів, коли слово зустрілося в корпусі – і так для кожного слова);

- далі масив слів сортується за частотою (слова зберігаються в геш-таблиці), і видаляються рідкісні слова;

- будується дерево Гаффмана, що часто застосовується для кодування словника – це значно знижує обчислювальну і часову складність алгоритму;

- з корпусу читається так зване субречення і проводиться субсемплювання слів з найбільшою частотою: субречення – це якийсь базовий елемент корпусу, зазвичай – просто речення, але це може бути і абзац, наприклад, або навіть ціла стаття, субсемплювання – це процес вилучення слів з найбільшою частотою з аналізу, що прискорює процес

навчання алгоритму і сприяє значному збільшенню якості отриманої в результаті моделі;

– за субреченням виконується прохід вікном (розмір вікна задається алгоритмом як параметр): в даному випадку під вікном мається на увазі максимальна дистанція між поточним словом і передбачуваним словом в реченні, тобто, якщо вікно дорівнює трьом, то для речення «Я зараз читаю книгу» аналіз (тобто застосування алгоритмів на базі однієї з архітектур Continuous Bag of Words або Skip-gram) буде проходити всередині блоку в три слова – для «Я зараз читаю», «зараз читаю книгу», вікно за замовчуванням дорівнює п'яти, рекомендованим значенням є десять;

– застосовується неймережа прямого поширення сигналу з функцією активації – ієрархічний софтмакс: в процесі навчання неймережа отримує вхідний вектор і намагається передбачити результат у вигляді розподілу ймовірностей слова бути в контексті вхідного слова на множині всіх слів, потім за допомогою функції втрат виконується штрафування моделі за неправильну класифікацію і нагородження за вірну [14]-[15].

Обчислення косинусної подібності виконується наступним чином на основі формул (2.1)-(2.2):

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}, \quad (2.1)$$

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_p^2}, \quad (2.2)$$

де $\text{sim}(x, y)$ – міра подібності між двома векторами ознак x і y ;

$\|x\|$ – Євклідова відстань;

$x_1, x_2, \dots, x_p$ – значення ознак [16].

Оцінка TF-IDF для слова t в документі d обчислюється множенням двох різних показників:

– частоти входження слова в документ: Існує кілька способів обчислення цієї частоти, причому найпростіший – це необроблена кількість

випадків, коли слово з'являється в документі, існують способи узгодження частоти з довжиною документа;

– зворотна частота входження слова у наборі документів визначає, наскільки поширене або рідкісне слово у всьому наборі документів: чим ближче воно до 0, тим більш поширене слово, цю метрику можна обчислити, взявши загальну кількість документів, розділивши її на кількість документів, що містять слово, і обчисливши логарифм, отже, якщо слово дуже поширене і зустрічається у багатьох документах, це число наблизиться до 0, в іншому випадку воно наблизиться до 1 [17].

Обчислення оцінки TF-IDF для оцінювання важливості слів у тексті відбувається на основі формул (2.3)-(2.5):

$$TF - IDF(t, d, D) = tf(t, d) \cdot idf(t, D), \quad (2.3)$$

$$tf(t, d) = \log(1 + freq(t, d)), \quad (2.4)$$

$$idf(t, D) = \log\left(\frac{N}{count(d \in D: t \in d)}\right), \quad (2.5)$$

де $freq(t, d)$ – частота появи слова t в документі d ;

N – загальна кількість документів;

$count(d \in D: t \in d)$ – кількість документів, в яких зустрічається слово t [17].

2.4 Запропонований метод рекомендації фільмів

Одним з механізмів, який найактивніше використовується для створення сучасних систем рекомендацій загалом є, штучні нейронні мережі. Саме їх було за результатами аналізу обрано для основи запропонованого методу.

Задача, що розглядається, є задачею прогнозування часового ряду, тобто за послідовністю вхідних ознак, що представляють собою множину значень $(m_1, m_2, \dots, m_n)$, необхідно виконати прогнозування вихідної ознаки,

що є рекомендацією для перегляду наступного фільму, m_{n+1} , на основі формули (2.6):

$$y = m_{n+1} = f(m_1, m_2, \dots, m_n), \quad (2.6)$$

де $m_1, m_2, \dots, m_n$ – уподобаний користувачем фільм під номером 1, 2, ..., n відповідно;

f – функціональна залежність між вхідними ознаками та вихідною ознакою;

y – вихідна ознака.

Для розв’язання цієї задачі було запропоновано наступний метод.

На першому етапі має бути реалізовано кодування даних. Мають бути закодовані уподобання користувачів. Це дозволить підвищити якість рекомендацій в результаті, оскільки якщо кожне уподобання характеризувати тільки ідентифікаційним номером фільму в базі даних, то це призводить до того, що невеликі неточності в прогнозі в результаті призводять до отримання номеру фільму, який може взагалі ніяк не відповідати смаку користувача. Тобто значний вплив на отримані результати має те, в якому порядку розташовані фільми в базі даних.

Виконання кодування даних дозволить зберігати в коді інформацію, невеликі зміни в якій мають з більшою ймовірністю характеризувати ближчі за різними характеристиками фільми.

Для кодування використовується метод word2vec. Навчання при цьому відбувається на основі описів фільмів. За результатами роботи методу створюються коди для кожного слова зі словника. Далі для кожного фільму за його описом виконується кодування, що ґрунтується на кодуванні слів. При цьому визначення коду опису виконується на основі оцінок TF-IDF, адже порівняно зі звичайним усередненням це дозволяє уникнути загальних слів, які не несуть впливу на сенс опису загалом (зокрема союзи, прийменники тощо). Код кожного слова множиться на його оцінку TF-IDF.

Отримане таким чином значення стає кодуванням фільму і представляє відповідне значення m_i .

На другому етапі має бути виконано підготовку даних. Під час підготовки даних зі всіх даних про уподобання користувачів має бути сформована навчальна вибірка, кожен екземпляр якої представлений наступним чином на основі формул (2.7)-(2.9):

$$s_{tr}^o = (x_o, y_o) = ((m_i, m_{i+1}, \dots, m_j), m_{j+1}), \quad (2.7)$$

$$(m_i, m_{i+1}, \dots, m_j) \subseteq S^u, \quad (2.8)$$

$$S^u = (m_1, m_2, \dots, m_k), \quad (2.9)$$

де $m_i, m_{i+1}, \dots, m_j, m_{j+1}$ – уподобаний користувачем фільм під номером $i, i+1, \dots, j, j+1$ відповідно;

S^u – множина уподобань фільмів користувача з номером u ;

s_{tr}^o – екземпляр навчальної вибірки з номером o ;

x_o – вхідні ознаки екземпляру навчальної вибірки з номером o ;

y_o – вихідна ознака екземпляру навчальної вибірки з номером o .

На третьому етапі має бути створено та навчено модель на основі ДКП.

На четвертому етапі має реалізовуватися рекомендація фільмів за поточними уподобаннями користувача.

Тобто для заданого набору останніх уподобань користувача за допомогою навченої моделі має бути виконано прогноз наступного уподобання користувача.

Далі для отриманого значення має бути виконане співставлення зі всіма фільмами, після чого обрано потрібну кількість найближчих фільмів, що і формують рекомендацію для перегляду фільмів.

2.5 Висновки за розділом 2

У розділі виконано аналіз методів рекомендації фільмів, серед яких: асоціативні правила, кластеризація, дерева рішень, штучні нейронні мережі, аналіз посилань, регресійний аналіз, байєсівські класифікатори, методи заповнення матриці. Виконано огляд принципів організації рекурентних нейронних мереж, зокрема довгої короткочасної пам'яті. Проаналізовано метод word2vec, який дозволяє виконувати кодування слів у вигляді вектору.

У розділі розроблено метод рекомендації фільмів, який характеризується виконанням рекомендації фільмів на основі використання довгої короткочасної пам'яті та використання в якості її вхідних ознак закодованих векторних представлень фільмів за допомогою використання методу word2vec, що дозволяє виконувати рекомендації фільмів з врахуванням послідовності уподобань користувача.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір засобів реалізації програмного забезпечення

В якості можливих варіантів для вибору мов програмування для реалізації програмного забезпечення було виділено мови програмування C#, Python, C++.

Основними перевагами мови програмування C# є:

- чиста об'єктно-орієнтована мова: будучи чисто об'єктно-орієнтованою мовою, C# дозволяє створювати код для зручного багаторазового використання та модульні сервісні програми;
- підтримка і інтеграція з переважною більшістю продуктів компанії Microsoft;
- цілісність та сумісність: застосунки, написані на .NET, мають кращу цілісність та сумісність з іншими .NET-технологіями, C# гарантує легку інтеграцію з компонентами, написаними різними мовами;
- безкоштовність ряду інструментів для невеликих компаній і деяких індивідуальних розробників – Visual Studio, хмара Azure, Windows Server, Parallels Desktop для Mac Pro тощо;
- типобезпечна мова: безпека типів в .NET запобігає потраплянню об'єктів одного типу в пам'ять інших об'єктів, ця функція не дозволяє розробнику робити помилки під час компіляції, а також під час роботи;
- типи даних мають фіксований розмір (32-бітний int і 64-бітний long), що підвищує «мобільність» мови і спрощує програмування;
- автоматичне «прибирання сміття» означає, що в більшості випадків не доведеться піклуватися про звільнення пам'яті: загальномовне середовище викликає збирач сміття і очищає пам'ять;
- велика кількість синтаксичного «цукру» – спеціальних конструкцій, розроблених для розуміння і написання коду: такі конструкції не мають значення при компіляції;

- низький поріг входження: синтаксис C# має багато схожого з іншими мовами програмування;

- багаті бібліотеки класів: наявність розширених бібліотек класів прискорює процес розробки та полегшує процес реалізації функцій [18]-[19].

До переваг мови програмування Python можна віднести наступні:

- інтерпретована мова: у випадку з інтерпретованими мовами код виконується безпосередньо рядок за рядком, якщо з'являється помилка, негайно повідомляється про неї та зупиняються подальші дії;

- легко вчити, читати та кодувати: причиною того, чому Python так часто рекомендують розробникам-новачкам, є його англоподібний синтаксис, процес вивчення, читання та написання коду вимагає набагато менше зусиль і рядків коду в порівнянні з іншими мовами;

- дуже продуктивна мова: підвищена швидкість роботи більшості застосунків та їх продуктивність обумовлені збільшеною потужністю управління, потужними функціями інтеграції та наявністю вбудованої можливості модульного тестування;

- величезна підтримка бібліотек: бібліотека Python настільки величезна, що можна знайти там більшість необхідних функцій (інструменти вебсервісу, Інтернет, рядкові операції тощо), якщо є потреба в чомусь іншому, існує понад 200 тисяч пакетів Python, які можна легко імпортувати за допомогою менеджера пакетів Python;

- портативний: якщо будуть включені функції, що залежать від системи, програму можна запускати на різних платформах без зміни коду;

- якісна і широка підтримка машинного навчання, що визначається як наявністю бібліотек, які повністю реалізують сучасні засоби глибокого навчання, створення штучних нейронних мереж, так і загалом простотою і гнучкістю мови, що дозволяє створювати такі рішення простіше [18].

У табл. 3.1 зведено результати порівняння мов програмування C++, C#, Python.

Таблиця 3.1 – Порівняння мов програмування

Критерій	C++	C#	Python
Клас мови	компільована, об'єктно-орієнтована (мультипарадигмальна)	компільована, чиста об'єктно-орієнтована (мультипарадигмальна)	інтерпретована, чиста об'єктно-орієнтована (мультипарадигмальна)
Можливості	низькорівневі, високорівневі	високорівневі	високорівневі
Типізація	статична	статична (динамічна)	динамічна
Створення програмних інтерфейсів	абстрактні класи	підтримується напрямую мовою	—
Створення Windows-застосунків	+	пряме призначення	можливе
Машинне навчання	можливе	можливе	розширене з широкою підтримкою

В результаті в якості засобів розробки було прийнято рішення використовувати наступні:

– мову програмування C# для створення Windows-застосунку з графічним інтерфейсом користувача, оскільки вона дозволяє швидше, простіше створювати Windows-застосунки, порівнюючи з найближчим аналогом – мовою програмування C++;

– мову програмування Python для створення засобів реалізації запропонованого методу, оскільки вона є більш зручною на сучасному етапі для організації машинного навчання.

Такий набір мов дозволяє використати для створення програмної системи переваги обох мов, при цьому, як було зазначено вище, мова програмування C# може зручно використовуватись з іншими мовами, а з іншого боку буде запропоновано таку структуру програмної системи, що дозволить працювати з різними компонентами, створеними різними мовами.

3.2 Функціональні вимоги до програмного забезпечення

Для забезпечення роботи користувача над даними фільмів та забезпечення його рекомендаціями щодо перегляду нових фільмів програмна система рекомендування фільмів повинна реалізовувати наступні функціональні вимоги:

- додавати та редагувати дані фільмів;
- вносити в систему та редагувати інформацію про акторів, режисерів, продюсерів, жанри;
- переглядати інформацію про фільми;
- виконувати пошук фільмів;
- додавати власні відгуки про фільми;
- переглядати всі відгуки про фільм;
- визначати середню оцінку фільму;
- додавати фільми до переліку для подальшого перегляду;
- переглядати та редагувати перелік для подальшого перегляду;
- оцінювати фільми;
- переглядати рейтинг найкращих фільмів за окремими жанрами або загалом;
- надавати рекомендації для перегляду наступних фільмів;
- виконувати авторизацію облікового запису;
- виконувати реєстрацію облікового запису.

Всі користувачі діляться на неавторизованих, авторизованих та адміністраторів.

На рис. 3.1 представлено діаграму прецедентів для роботи відповідних типів користувачів з програмною системою за описаними функціональними вимогами.



Рисунок 3.1 – Діаграма прецедентів

3.3 Проектування бази даних

Для збереження даних про фільми необхідно було створити базу даних.

Системою керування базами даних (СКБД) для цієї бази даних було обрано MySQL [20]-[21].

За результатами аналізу предметної області було виділено наступні сутності:

- film – фільм;
- rate – оцінка;
- user – користувач;
- person – особа (режисер або продюсер);
- actor – актор;
- comment – коментар;
- recommendation – рекомендація для користувача для перегляду

фільмів;

- genre – жанр.

Таблиця film має наступні поля:

- номер id;
- постер fimage;
- назва фільму title;
- опис description;
- слоган motto;
- бюджет budget;
- всесвітні збори fees;
- вікові обмеження agelevel;
- час (тривалість фільму) duration;
- країна-виробник або країни-виробники devcount;
- дата прем'єри у світі dataprem;
- середня оцінка фільму avscore.

Таблиця rate має наступні поля:

- номер id;
- оцінка фільму score;
- фільм film_id;
- автор оцінки user_id;
- дата і час виставлення оцінки dtime.

Таблиця user має наступні поля:

- номер id;
- ім'я name;
- прізвище surname;
- логін login;
- пароль password;
- зображення photo;
- роль role (адміністратор або користувач).

Таблиця person має наступні поля:

- номер id;
- ім'я name;
- прізвище surname.

Таблиця actor має наступні поля:

- номер id;
- роль role;
- ім'я та прізвище surname;
- фото photo.

Таблиця comment має наступні поля:

- номер id;
- текст ctext;
- автор коментаря user_id;
- фільм film_id;
- дата і час коментування dtime.

Таблиця genre має наступні поля:

- номер id;
- назва name.

Таблиця recommendation має наступні поля:

- номер id;
- фільм film_id;
- отримувач рекомендації user_id;

– час, до якого буде відображатися (буде активною) рекомендація, –
dtime.

Таблиця actor_film містить наступні поля:

- номер id;
- актор actor_id;
- фільм film_id.

Для реалізації зв'язків багато-до-багатьох було створено наступні таблиці:

- actor_film – актор фільму (зв'язок між таблицями actor та film);
- producer_film – продюсер фільму (зв'язок між таблицями person та film);
- director_film – режисер фільму (зв'язок між таблицями person та film);
- film_genre – жанр фільму (зв'язок між таблицями genre та film).

Таблиця producer_film містить наступні поля:

- номер id;
- продюсер person_id;
- фільм film_id.

Таблиця director_film містить наступні поля:

- номер id;
- режисер person_id;
- фільм film_id.

Таблиця film_genre містить наступні поля:

- номер id;
- жанр genre_id;
- фільм film_id.

На рис. 3.2 представлено відповідну схему бази даних.

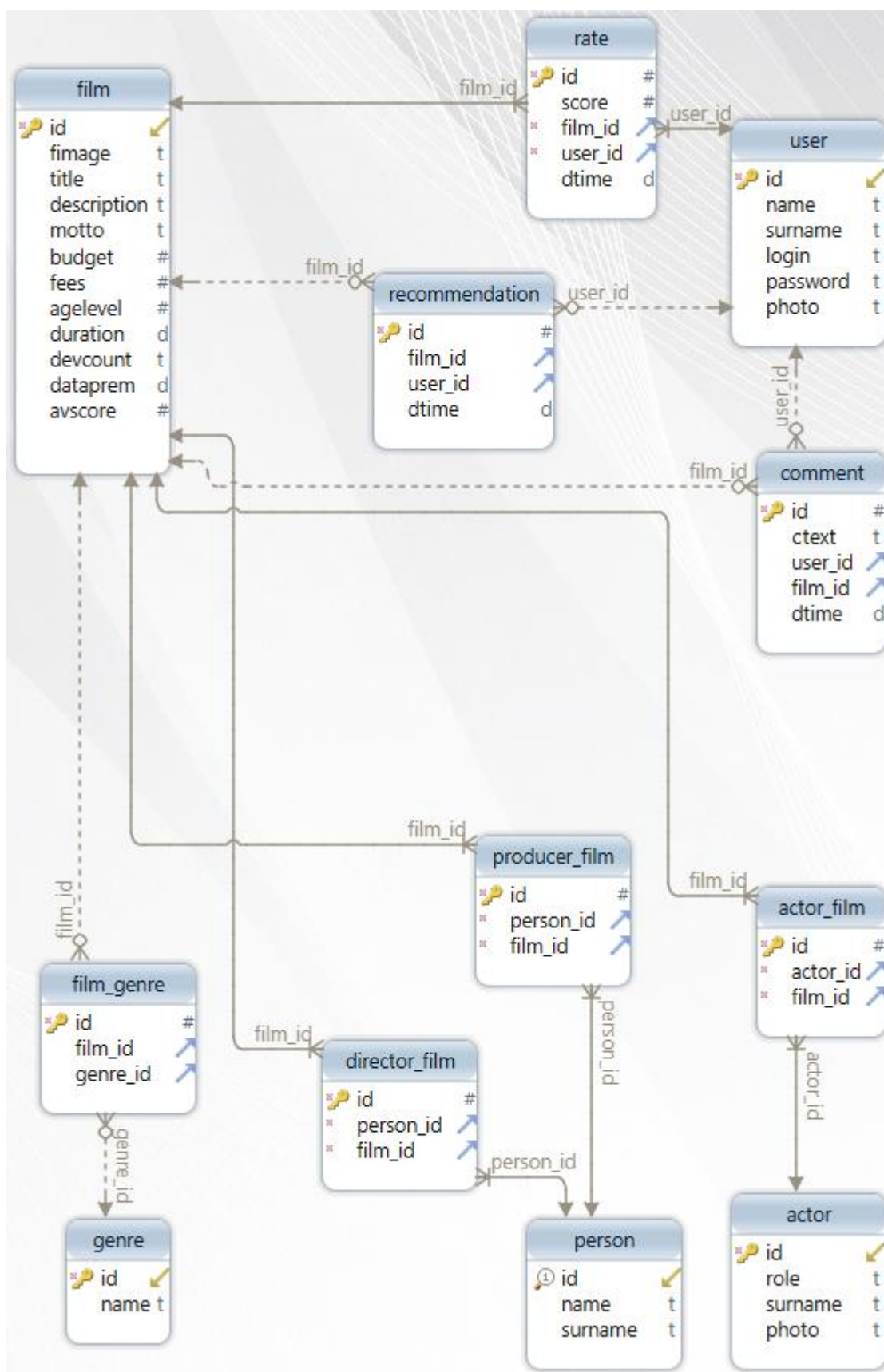


Рисунок 3.2 – Схема бази даних

3.4 Структура програмного забезпечення

Структурну схему програмної системи представлено на рис. 3.3.



Рисунок 3.3 – Структурна схема програмної системи

Програмне забезпечення має бути розроблено у вигляді програмної системи, що включає наступні складові:

- підсистема для організації моделі предметної області;
- підсистема графічного інтерфейсу користувача;
- підсистема рекомендації фільмів;
- база даних.

Така логіка організації дозволяє взаємодіяти частині, що відповідає за створення безпосередньо Windows-застосунку, з частиною, що відповідає за забезпечення рекомендації фільмів, через базу даних. Це є єдиною точкою накопичення даних, з якої отримуються та вносяться дані в потрібний момент.

Підсистема для організації моделі предметної області дозволяє перетворити дані з бази даних на об'єктно-орієнтовану форму і працювати в подальшому з графічним інтерфейсом. Відповідно дана підсистема має складатися передовсім з програмних файлів, кожен з яких відповідає конкретній сутності предметної області.

Підсистема графічного інтерфейса користувача повинна забезпечувати роботу форм у відповідності з функціональними вимогами, які було описано вище.

3.5 Висновки за розділом 3

Обрано мову програмування C# для створення Windows-застосунку з графічним інтерфейсом користувача та мову програмування Python для створення засобів реалізації запропонованого методу.

Визначено функціональні вимоги до розроблюваного програмного забезпечення. Представлено діаграму прецедентів для роботи неавторизованого, авторизованого користувачів та адміністратора з системою.

Обрано MySQL в якості СКБД для проектування та реалізації бази даних. Виконано проектування бази даних з представленням відповідної схеми бази даних.

Визначено структуру програмної системи, що складається з підсистеми для організації моделі предметної області, підсистеми графічного інтерфейса користувача, підсистеми рекомендації фільмів та бази даних.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Алгоритм функціонування системи

Функціонування програмної системи організовано на основі отримання щодобового переліку рекомендацій для користувачів, який кожен день використовується в роботі програмного меню, яке надає доступ до всіх функцій застосунку.

Відповідно робота програми розпочинається з завантаження даних з бази даних. На основі цих даних відбувається кодування даних, їх підготовка, після чого реалізується створення і навчання моделі. Якщо така навчена модель вже існує, то вона завантажується в пам'ять.

Далі реалізується потенційно нескінченний цикл. На кожній його ітерації забезпечується щоденне повторення рекомендування та нескінченного циклу, в межах якого користувач взаємодіє з програмою.

Рекомендування реалізується наступним чином. Спочатку витягаються дані з бази даних. Після цього за рахунок навченої моделі виконується формування рекомендацій для кожного користувача з бази даних.

Нескінченний цикл користувацького меню передбачає звернення користувача для виклику різних функціональних можливостей. В межах цього циклу зокрема відбувається виведення рекомендацій фільмів для користувача. Оновлення рекомендацій виконується під час звернення до бази даних і витягання нових значень. Окрім того користувач виконує оцінювання фільмів, на основі яких в подальшому, наступного дня, буде сформовано наступні рекомендації фільмів.

На рис. 4.1 представлено відповідну описану функціональну схему програми.

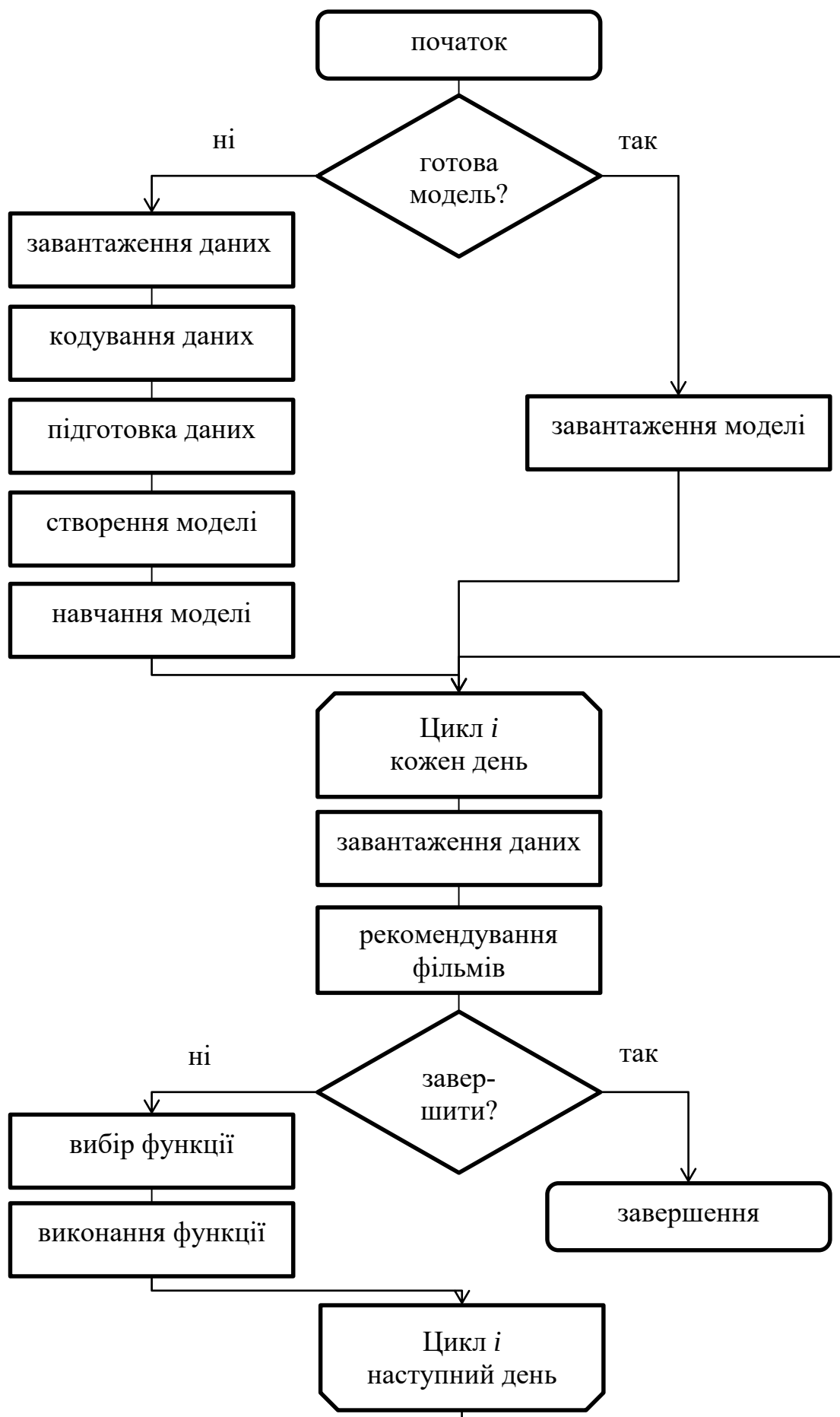


Рисунок 4.1 – Функціональна схема програми

4.2 Опис модулів реалізованої системи

Для підсистеми для організації моделі предметної області було реалізовано наступні модулі:

- Actor.cs – реалізує клас актора;
- Comment.cs – реалізує клас коментаря;
- Film.cs – реалізує клас фільму;
- Genre.cs – реалізує клас жанру;
- Person.cs – реалізує клас особи;
- Rate.cs – реалізує клас оцінки;
- Recommendation.cs – реалізує клас рекомендацій;
- User.cs – реалізує клас користувача.

До прикладу розглянемо побудову найголовнішої сутності предметної області, за яку відповідає клас Film. Усі інші класи побудовано за подібною аналогією.

Клас Film містить наступні поля:

- int id – ідентифікаційний номер;
- string title – назва фільму;
- string dataprem – дата прем'єри у світі;
- double budget – бюджет;
- string devcount – країна-виробник або країни-виробники;
- string motto – слоган;
- string[] description – опис;
- double fees – всесвітні збори;
- double duration – тривалість фільму;
- string agelevel – вікові обмеження;
- string fimage – постер;
- double avscore – середня оцінка фільму;
- List<Genre> genres = new List<Genre>() – перелік жанрів фільму,

кожен з яких є об'єктом класу Genre;

- `List<Actor> actors = new List<Actor>()` – перелік акторів фільму, що об'єктами класу `Actor`;
- `List<Person> producers = new List<Person>()` – перелік продюсерів фільму, що об'єктами класу `Person`;
- `List<Person> directors = new List<Person>()` – перелік режисерів фільму, що об'єктами класу `Person`;
- `List<Rate> scores = new List<Rate>()` – перелік оцінок, виставлених фільму, що об'єктами класу `Rate`;
- `List<Comment> comments = new List<Comment>()` – перелік коментарів до фільму, що об'єктами класу `Comment`.

Для кожного поля передбачено метод для отримання значення поля та встановлення його в нове значення. Окрім того забезпечено конструктор без параметрів `Film()`, який встановлює значення за замовчуванням для кожного поля.

У складі цієї підсистеми організовано також клас `Database`, кожен метод якого реалізує запити до бази даних на вибірку даних, оновлення даних, вставку нових даних.

Для організації підсистеми графічного інтерфейсу користувача створено зокрема наступні класи відповідних форм:

- `FilmRecomAuth.cs` – форма авторизації, яка в розширеному варіанті містить засоби реєстрації;
- `FilmRecomAdmin.cs` – основна форма роботи адміністратора;
- `FilmRecomAdminFilm.cs` – форма роботи адміністратора з окремим фільмом (створення або редагування);
- `FilmRecomUser.cs` – основна форма роботи користувача;
- `FilmRecomUserOne.cs` – форма перегляду даних про фільм користувачем;
- `FilmRecomComments.cs` – форма роботи з відгуками про фільм.

Клас `FilmRecomAdminFilm` має наступні поля:

- `Database db` – об'єкт, що представляє підключення до бази даних;

- List<Film> AllFilms – перелік всіх фільмів;
- CurrFilm – поточний номер фільму (необхідний оскільки форма окрім роботи з одним фільмом дозволяє переходити до наступних, попередніх фільмів у переліку);
- CurrActor – поточний номер актору;
- CurrProd – поточний номер продюсера;
- CurrDir – поточний номер режисера;
- IMPATH – шлях до теки з зображеннями, які використовуються в якості постерів фільмів.

Клас FilmRecomAdminFilm надає доступ зокрема до наступних методів:

- FilmRecomAdminFilm() – конструктор, який забезпечує ініціалізацію компонентів форми та визначення параметрів її відображення;
- void NewFilm() – метод, який виконує створення нового об'єкту фільму, додавання його до переліку та оновлення номеру;
- void GetFilm() – метод, який виконує заповнення даних про поточний фільм з переліку фільмів;
- void NewProd() – метод, який виконує створення нового об'єкту продюсера, додавання його до переліку та оновлення номеру;
- void GetProducers() – метод, який виконує заповнення даних про поточного продюсера з переліку фільмів;
- void NewDir() – метод, який виконує створення нового об'єкту режисера, додавання його до переліку та оновлення номеру;
- void GetDirectors() – метод, який виконує заповнення даних про поточного режисера з переліку фільмів;
- void NewActor() – метод, який виконує створення нового об'єкту актора, додавання його до переліку та оновлення номеру;
- void GetActors() – метод, який виконує заповнення даних про поточного актора з переліку фільмів;

- void GetAllData() – метод, який реалізує заповнення переліку фільмів даними, завантаженими з бази даних;
- void FreeFilm() – метод, який забезпечує очищення даних про фільм;
- void SaveData() – метод, який реалізує збереження в базу даних даних поточного переліку фільмів у відповідності з виконаними на поточний момент змінами.

Окрім того реалізовано набір методів для оброблення подій, пов'язаних з графічними елементами, зокрема:

- void btnSetClick(object sender, EventArgs e) – метод, який забезпечує оброблення натискання на кнопку збереження результатів створення нового фільму;

- void btnBeforeClick(object sender, EventArgs e) – метод, який виконує перехід до попереднього фільму в переліку, що за необхідності дозволяє виконувати редагування іншого фільму;

- void tbMottoTChang(object sender, EventArgs e) – метод, який забезпечує редагування слогану фільму в переліку у відповідності зі зміною в інтерфейсі або збереження його у випадку введення першого значення при додаванні;

- void tbTitleTChang(object sender, EventArgs e) – метод, який забезпечує фіксацію встановленої в формі назви фільму у відповідній позиції переліку фільмів.

Аналогічно організовано інші обробники подій, а також загалом інші перелічені форми.

Для реалізації запропонованого методу створено модуль мовою програмування Python FilmRecomMethod.py.

У даному модулі реалізовано 2 моделі.

Перша модель використовується для кодування і її створення відбувається на основі моделі Word2Vec:

```
WVmodel = gensim.models.Word2Vec (alltext, min_count = 2, size =
100, window = 5)
```

Вказані параметри використовуються під час навчання моделі: `min_count` визначає мінімальну кількість разів, яку має зустрітись слово, щоб його було враховано в побудованій моделі, `size` – розмірність результуючого вектора, `window` – розмір вікна, що визначає кількість слів до поточного та після поточного слова, які буде враховано в якості контексту для прогнозування.

Друга модель побудована на основі ДКП.

Створення моделі реалізується наступним чином:

```
LSTMmodel = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(128, return_sequences=True),
    tf.keras.layers.LSTM(128, return_sequences=True),
    tf.keras.layers.Dense(32, activation="relu"),
    tf.keras.layers.Dense(units=1)
])
```

Компіляція та навчання моделі забезпечується наступним викликом:

```
LSTMmodel.compile(loss=tf.keras.losses.MSE(),
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.1),
    metrics=["accuracy"])
res = LSTMmodel.fit(trdata, epochs=1000)
```

Отримання рекомендованого фільму за ланцюжком останніх уподобань реалізується наступним чином:

```
rf = LSTMmodel.predict(lastlikedfilms)
```

4.3 Висновки за розділом 4

Розглянуто процес функціонування програмного забезпечення з наведенням функціональної схеми. Описано реалізовані модулі та принципи організації основних реалізованих модулів у межах програмної системи.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

5.1 Призначення програми

Програмне забезпечення рекомендації фільмів має використовуватись для отримання широкої інформації про випущені в світі фільми, створення власного переліку для подальшого перегляду та отримання рекомендацій для подальшого перегляду.

5.2 Умови виконання програми

Для роботи програмного забезпечення необхідно на серверній частині інсталиювати:

- інтерпретатор мови програмування Python версії 3;
- середовище Microsoft Visual Studio 2019;
- систему керування базами даних MySQL;
- пакет tensorflow.

Для експлуатації програмного забезпечення необхідні наступні технічні засоби на серверній частині:

- процесор з 4 ядрами, при цьому частота кожного має бути не менше 3 Гц;
- оперативна пам'ять – не менше 4 Гб;
- жорсткий диск – вільний простір не менше 2 Гб.

5.3 Виконання програми

Після запуску програми (виконуваного файлу FR.exe) на екрані з'являється вікно авторизації акаунту. Необхідно ввести власні дані або зареєструвати свій акаунт. Після успішної авторизації, якщо акаунт є користувацьким, то відображається основна форма роботи користувача (рис. 5.1).

Дослідження та програмна реалізація методів рекомендування фільмів

рекомендації watchlist вийти

кращі жанр всі

знайти

Постер	Назва	Рік	Країна	Жанр	Оцінка	Вік
	Хрещений батько	1972	США	Драма	9,1	R
	Інтерстеллар	2014	США, Велика Британія	Драма, Трилер	8,4	16
	Лабіринт Фавна	2006	Мексика, Іспанія, США	Драма, Фантастика	8,5	16
	Гладіатор	2000	США, Велика Британія	Бойовик, Драма, Пригоди	8,1	R
	Амадей	1984	США	Драма, Історичний	7,1	R
						

Рисунок 5.1 – Основна користувацька форма

Центральним елементом основної користувацької форми (так само і адміністраторської) є таблиця, в яку виводиться перелік фільмів.

Можна натиснути на будь-який елемент таблиці і відкриється вікно з формою детального перегляду даних фільму (рис. 5.2).

Дана форма дозволяє переглянути постер, значення основних параметрів фільму, візуально визначену (набором зірок) оцінку фільму, а також фактичне значення оцінки, переглянути повний опис фільму, що складається з українського та англійського (використовується для кодування під час роботи запропонованого методу) варіанту. Окрім того видається перелік акторів з їх фотографіями.

Окрім того надається можливість переглянути відгуки фільму (для цього передбачена кнопка «відгуки») та додати фільм до переляку фільмів користувача для подальшого перегляду (кнопка «додати у watchlist»).

На рис. 5.3 представлено форму перегляду відгуків користувачів про фільм.

Дослідження та програмна реалізація методів рекомендації фільмів



Втеча з Шоушенка (1994) ★★★★★★ 9,2

слоган "Страх - це кайдани. Надія - це свобода."

виробництво США

жанр Драма

дата прем'єри 25.09.1994

режисер Френк Дарабонт

продюсер Нікі Марвін

тривалість 142 хв

віковий рейтинг R

бюджет 25 000 000 \$

збори 28 000 000 \$

додати у watchlist

відгуки

Хроніки досвіду колишнього успішного банкіра в'язнем в похмурій тюрмі Шоушенка після визнання його винним у злочині, якого він не вчинив. Фільм змальовує унікальну манеру спілкуватися з новим, мучительним життям; по дорозі він дружить із кількома ув'язненими, особливо мудрим довгостроковим ув'язненим на ім'я Ред. Chronicles the experiences of a formerly successful banker as a prisoner in the gloomy jailhouse of Shawshank after being found guilty of a crime he did not commit. The film portrays the man's unique way of dealing with his new, torturous life; along the way he befriends a number of fellow prisoners, most notably a wise long-term inmate named Red.

Енді Дюфрейн	Ред	Директор
		
Тім Роббінс Депп	Морган Фрімен Блум	Боб Гантон Найтлі

Рисунок 5.2 – Перегляд даних фільму

Дослідження та програмна реалізація методів рекомендації фільмів

Темні часи

25 листопада 2020 р. ____

ravlo
Оцінка : 9
Супер!

26 листопада 2020 р. ____

tetro
Оцінка : 5
Важливо розуміти, що це як не як художній фільм

26 листопада 2020 р. ____

user678
Оцінка : 8
Гарний, повчальний

Ваш відгук та оцінка ★★★★★★★★★★

Відправити

Рисунок 5.3 – Відгуки користувачів про фільм

Форма перегляду відгуків дозволяє також додати власний відгук про фільм та встановити оцінку фільму (це можна зробити тільки на даній формі).

Якщо на основній користувацькій формі ввести в поле пошуку назву фільму або її частину і натиснути на кнопку «знайти», то в таблиці нижче буде відображено фільми, які відповідають за назвою введеному значенню (рис. 5.4).

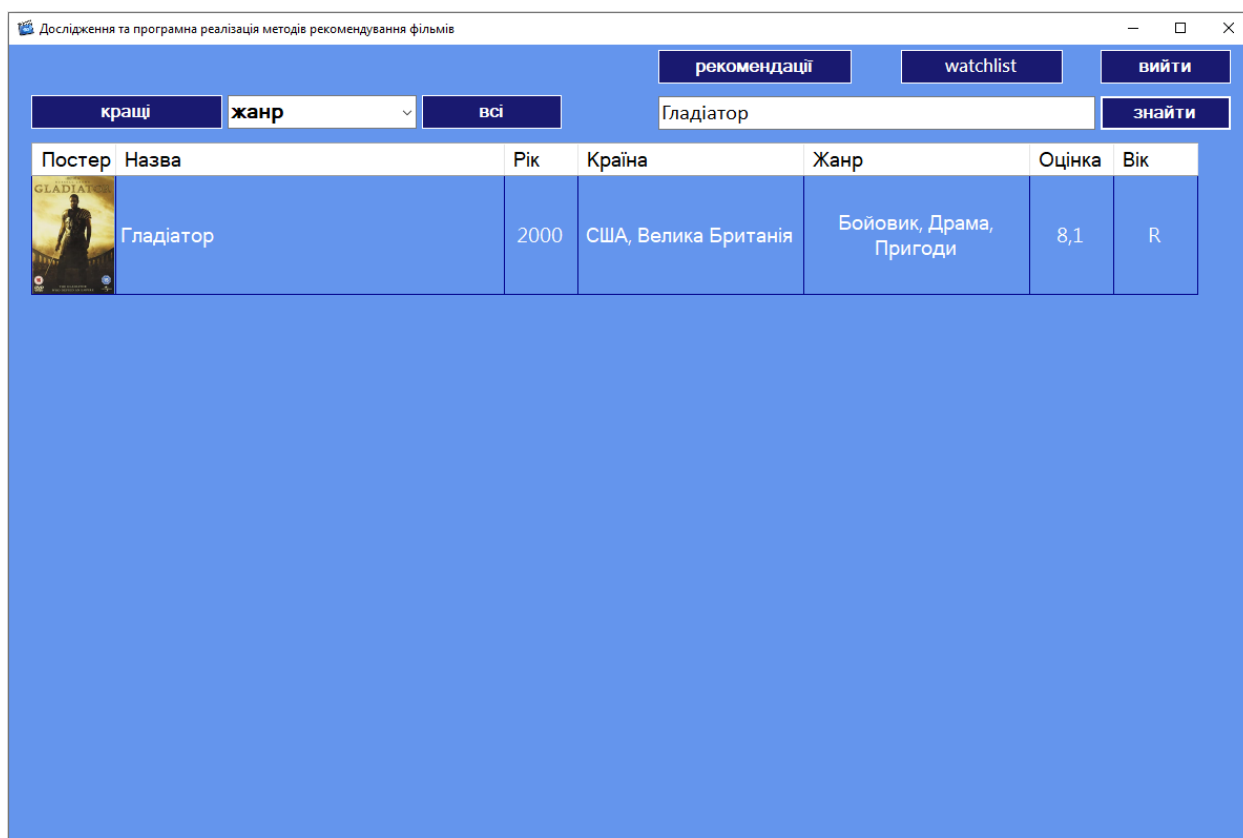


Рисунок 5.4 – Результати пошуку фільмів

Якщо на основній користувацькій формі натиснути на кнопку «watchlist», то в таблиці буде відображено перелік фільмів для перегляду користувачем (рис. 5.5).

Якщо обрати один з фільмів, то відкриється вікно перегляду інформації про фільм, де міститься кнопка «вилучити з watchlist», що дозволяє вилучити фільм з переліку (рис. 5.6).

Дослідження та програмна реалізація методів рекомендування фільмів

рекомендації watchlist вийти

кращі жанр всі

знайти

Постер	Назва	Рік	Країна	Жанр	Оцінка	Вік
	Рома	2018	Мексика	Драма	7,7	R
	Готель Гранд Будапешт	2014	Німеччина, США	Пригоди, Комедія	7,2	R
	Зелена книга	2018	США	Драма, Комедія	8,33	12

Рисунок 5.5 – Перелік фільмів для перегляду користувачем

Дослідження та програмна реалізація методів рекомендування фільмів



Зелена книга (2018)

★★★★★ 8,33

слоган "Надихаючись справжньою дружбою"

виробництво США

жанр Драма, Комедія

дата прем'єри 18.11.2018

режисер Пітер Фарреллі

продюсер Джим Берк, Браян Г'юес Каррі, Пітер Фарреллі

тривалість 130 хв

віковий рейтинг 12

бюджет 23 000 000 \$

збори 85 000 000 \$

вилучити з watchlist

відгуки

Італійсько-американський вибивач з робочого класу стає водієм класичного афроамериканського піаніста під час туру по майданчиках американським Півднем 1960-х.
A working-class Italian-American bouncer becomes the driver of an African-American classical pianist on a tour of venues through the 1960s American South.

Тоні Валлелонга



Біро Мортенсен

Дон Ширлі



Магершала Алі Суїнтон

Долорес



Лінда Карделліні

Жуль Поделл



Дон Старк Вонг

Джонні Венере



Себастьян Маніскалко

Рисунок 5.6 – Перегляд даних фільму зі списку перегляду користувача

Якщо на основній користувацькій формі натиснути на кнопку «кращі», то в таблиці буде відображено перелік найкращих фільмів за рейтингом, тобто за зменшенням користувацької оцінки (рис. 5.7).

Постер	Назва	Рік	Країна	Жанр	Оцінка	Вік
	Темний лицар	2008	США, Велика Британія	Бойовик, Драма, Трилер	9,3	PG-13
	Втеча з Шоушенка	1994	США	Драма	9,2	R
	Хрещений батько	1972	США	Драма	9,1	R
	Бійцівський клуб	1999	США	Драма	9,1	16
	Міцний горішок	1988	США	Бойовик, Трилер	9	R

Рисунок 5.7 – Перегляд переліку найкращих фільмів

Якщо з випадаючого переліку «жанр» обрати один з жанрів, то буде виведено перелік найкращих за рейтингом фільмів, які відповідають вказаному жанру (рис. 5.8).

Для того щоб знову заповнити таблицю всіма наявними фільмами, необхідно натиснути на кнопку «всі».

Якщо натиснути на кнопку «рекомендації» то відкриється форма, на якій буде представлено інформацію про фільм, який рекомендовано користувачу для перегляду. Прогнозування виконувалось за останніми 8 уподобаними фільмами користувачем. До уподобаних належать фільми, яким користувач виставив оцінку вище 7. Рекомендацією є наступний фільм (рис. 5.9).

Дослідження та програмна реалізація методів рекомендування фільмів

рекомендації watchlist вийти

кращі Драма всі знайти

Постер	Назва	Рік	Країна	Жанр	Оцінка	Вік
	Темний лицар	2008	США, Велика Британія	Бойовик, Драма, Трилер	9,3	PG-13
	Втеча з Шоушенка	1994	США	Драма	9,2	R
	Бійцівський клуб	1999	США	Драма	9,1	16
	Хрещений батько	1972	США	Драма	9,1	R
	Список Шиндлера	1993	США	Драма, Історичний	8,9	16

Рисунок 5.8 – Перегляд переліку найкращих фільмів за жанром

Дослідження та програмна реалізація методів рекомендування фільмів

Рекомендуємо переглянути наступним

Ірландець (2019)

★★★★★ 7,8

слоган "Його історія змінила історію"

виробництво США

жанр Драма

дата прем'єри 27.11.2019

режисер Мартін Скорсезе

продюсер Мартін Скорсезе, Роберт Де Ніро, Джейн Розенталь, Гастон Павлович, Емма Коскофф

тривалість 209 хв

віковий рейтинг R

бюджет 159 000 000 \$

збори 1 000 000 \$

додати у watchlist

відгуки

Френк "Ірландець" Ширан - людина, у якої багато на умі. Колишній вищий чиновник профспілки та вбивця навчився вбивати під час служби в Італії протягом Другої світової війни. Тепер він озирається на своє життя та вбивства, що визначали його мафіозну кар'єру, підтримуючи зв'язок із злочинною родиною Буфаліно. Зокрема, роль, яку він зіграв у зникненні свого друга життя Джиммі Гоффа, колишнього президента Міжнародного братства водіїв вантажівок, який загадково зник наприкінці липня 1975 року у віці 62 років.

Frank "The Irishman" Sheeran is a man with a lot on his mind. The former labor union high official and hitman, learned to kill serving in Italy during the Second World War. He now looks back on his life and

Френк «Ірландець» Роберт Де Ніро Фокс

Джиммі Хоффа Аль Пачіно Ллойд

Рассел Буфаліно Джо Пеші Томпсон

Рисунок 5.9 – Перегляд рекомендацій для наступного перегляду

Для того щоб вийти з авторизованого акаунту, потрібно натиснути на кнопку «вийти». Тоді знову буде відкрито форму авторизації.

Неавторизованому користувачу відповідно недоступні кнопки «рекомендації», «watchlist», а на формі відгуків присутній тільки набір відгуків, а додавати власний відгук та виставляти власну оцінку неможливо.

Адміністратору після авторизації доступна адміністраторська форма, аналогічна користувацькій, але при цьому можна або редагувати фільми, або виконувати їх пошук, або додавати нові.

Форма додавання фільму представлена на рис. 5.10.

Рисунок 5.10 – Додавання фільму

Для додавання акторів, режисерів та продюсерів потрібно або ввести ім'я, після чого натиснути на кнопку «шукати» і за наявності буде виведено зображення актора з бази даних, або потрібно ввести дані актору, після чого натиснути на кнопку «додати» і запис буде створений, а актор буде доданий до фільму.

Кнопка «обрати» на панелі акторів дозволяє обрати потрібне фото. Кнопка «обрати» на загальній формі дозволяє аналогічним чином обрати постер до фільму.

Перелік «Жанр» дозволяє обирати жанри фільму, після чого правіше текст через кому буде поповнюватися доданими жанрами (рис. 5.11).

Рисунок 5.11 – Внесення даних фільму

На рис. 5.12 продемонстровано форму редагування даних про фільм.

За наявності акторів, продюсерів або режисерів (за відповідними вкладками) у фільмі, можна користуватися кнопками «назад» і «вперед» для переміщення між записами акторів, продюсерів або режисерів.

Кнопки «попередній» і «наступний» дозволяють переміщатися між фільмами послідовно. Кнопка «зберегти» дозволяє зберегти результати додавання або редагування фільму.

Програма обробляє помилки, виявлені під час роботи, і реагує на них виведенням повідомлень (рис. 5.13).

Дослідження та програмна реалізація методів рекомендації фільмів



Назва: Темні часи

Слоган: Ніколи не здавайся!

Виробництво: Велика Британія

Жанр: Трилер, Драма, Історичний, Трилер

Дата прем'єри: 26.11.2017

Тривалість (хвилин): 125

Вікова категорія: PG-13

Бюджет (млн.): 30

Касові збори (млн.): 150,8

Постер

обрати

Опис

Під час Другої світової війни, коли потужний вермахт Адольфа Гітлера бушує по всій Європі, прем'єр-міністр Великої Британії Невіл Чемберлен змушений подати у відставку, рекомендує Вінстона Черчілля як свою заміну. Але навіть у перші дні свого перебування лідером країни, на Черчілля чиниться тиск для того, щоб розпочати мирні переговори з Гітлером або боротися напівпрямую зі, здавалося б, непереможною нацистською владою, якою б не була ціна. Яким би складним і небезпечним не було його рішення, Черчіллю нічого не залишається, як блищати в найтемнішу годину країни. During World War II, as Adolf Hitler's powerful Wehrmacht rampages across Europe, the Prime Minister of the United Kingdom, Neville Chamberlain, is forced to resign, recommending Winston Churchill as his replacement. But even in his early days as the country's leader, Churchill is under pressure to commence peace negotiations with Hitler or to fight head-on the seemingly invincible Nazi regime,

Актори

Режисери

Продюсери

Ім'я: Джо

Прізвище: Райт

додати

шукати

назад

вперед

новий

попередній

наступний

зберегти

Рисунок 5.12 – Редагування даних фільму

Дослідження та програмна реалізація методів рекомендації фільмів



Назва: Темні часи

Слоган: Ніколи не здавайся!

Виробництво: Велика Британія

Жанр: Трилер, Драма, Історичний, Трилер

Дата прем'єри: 26.11.2017

Тривалість (хвилин): 125

Вікова категорія: PG-13

Бюджет (млн.): 30

Касові збори (млн.): 150,8

Постер

обрати

Опис

Під час Другої світової війни, коли потужний вермахт Адольфа Гітлера бушує по всій Європі, прем'єр-міністр Великої Британії Невіл Чемберлен змушений подати у відставку, рекомендує Вінстона Черчілля як свою заміну. Але навіть у перші дні свого перебування лідером країни, на Черчілля чиниться тиск для того, щоб розпочати мирні переговори з Гітлером або боротися напівпрямую зі, здавалося б, непереможною нацистською владою, якою б не була ціна. Яким би складним і небезпечним не було його рішення, Черчіллю нічого не залишається, як блищати в найтемнішу годину країни. During World War II, as Adolf Hitler's powerful Wehrmacht rampages across Europe, the Prime Minister of the United Kingdom, Neville Chamberlain, is forced to resign, recommending Winston Churchill as his replacement. But even in his early days as the country's leader, Churchill is under pressure to commence peace negotiations with Hitler or to fight head-on the seemingly invincible Nazi regime,

Актори

Режисери

Продюсери

Ім'я: Джо

Прізвище: Райт

додати

шукати

назад

вперед

новий

попередній

наступний

зберегти

Рисунок 5.13 – Повідомлення про успішне внесення даних

Якщо фільм успішно збережено в базі даних, то виводиться повідомлення про це. Якщо дану дію не вдалося виконати, то також виводиться повідомлення.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

Проект програмного забезпечення, що розробляється, призначений для підбору кінофільмів. Даний програмний додаток дозволить швидко аналізувати та оброблювати великий масив даних. Створений алгоритм може бути використаний як у якості складової частини розроблюваного додатку, так і може бути доопрацьований для роботи з іншими ресурсами аналогічного спрямування.

Економічна вигода – немає необхідності наймати експертів для формування переліку рекомендованих фільмів, який до того ж є неперсоналізованим.

Програмне забезпечення дозволяє підвищити продуктивність праці (процес роботи експерта виконується за 200 години замість попередніх 500 годин, тобто він реалізує лише загальний нагляд), що призводить до економії заробітної плати експерта, яка складає 90 грн. на годину.

6.1 Планування розробки програмного продукту

Для розрахунку витрат необхідно визначити етапи робіт та їх тривалість. У розробці беруть участь програміст протягом 3,5 місяців та консультант протягом 0,25 місяця. Розробка програми починається 1 вересня та повинна бути закінчена до 13 грудня 2020 року.

Тривалість робіт визначають за формулою (6.1):

$$T_{\text{ц}} = \frac{Q}{R}, \quad (6.1)$$

де $T_{\text{ц}}$ – тривалість циклу, днів;

Q – трудомісткість, людино-днів;

R – кількість виконавців, чол.

Отримана інформація зведена в табл. 6.1.

Таблиця 6.1 – Завдання та обов'язки по розробці програми

Назва роботи	Трудомісткість		Виконавці	Тривалість, днів
	люд.- дні	% до підсумку		
1. Аналіз предметної області	2	2,56	Програміст	2
2. Узгодження результатів аналізу	6	7,69	Програміст Консультант	3
3. Визначення вимог до програмного продукту	4	5,12	Програміст	4
4. Узгодження вимог	4	5,12	Програміст Консультант	2
5. Документування вимог	3	3,85	Програміст	3
6. Проектування структури програми	19	24,36	Програміст	19
7. Розробка схеми функціонування програми	5	6,41	Програміст	5
8. Створення програмного коду	18	23,08	Програміст	18
9. Тестування та налагодження програмного продукту	5	6,41	Програміст	5
10. Складання програмної документації	12	15,4	Програміст	12
Разом	78	100		73

За даними табл. 6.1 складається зведений стрічковий графік

планування розробки програмного продукту (табл. 6.2).

Таблиця 6.2 – Зведений стрічковий графік планування розробки програмного застосунку (частина 1)

Найменування роботи	Календарний період, дні														
	1.09-6.09	7.09-13.09	14.09-20.09	21.09-27.09	28.09-4.10	5.10-11.10	12.10-18.10	19.10-25.10	26.10-1.11	2.11-8.11	9.11-15.11	16.11-22.11	23.11-29.11	30.11-6.12	7.12-13.12
1. Аналіз предметної області															
2. Узгодження результатів аналізу															
3. Визначення вимог до програмного продукту															
4. Узгодження вимог															
5. Документування вимог															
6. Проєктування структури програми															
7. Розробка схеми функціонування програми															
8. Створення програмного коду															

Продовження графіку приведено в табл. 6.3.

Таблиця 6.3 – Зведений стрічковий графік планування розробки програмного застосунку (частина 2)

Найменування роботи	Календарний період, дні													
	1.09-6.09	7.09-13.09	14.09-20.09	21.09-27.09	28.09-4.10	5.10-11.10	12.10-18.10	19.10-25.10	26.10-1.11	2.11-8.11	9.11-15.11	16.11-22.11	23.11-29.11	30.11-6.12
9. Тестування та налагодження програмного продукту														
10. Складання програмної документації														

Стрічковий графік планування розробки програмного продукту представляє собою таблицю, в першому стовпці якої розміщені в порядку збільшення термінів початку виконання всі види роботи, а за другою координатою – календарний період їх виконання [22]. Чорним кольором приведено роботу програміста, а сірим – консультанта.

6.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складається калькуляція кошторисної вартості робіт, яка включає наступні статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок;

- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- накладні витрати;
- податки [22].

6.2.1 Розрахунок основної заробітної плати

Витрати за цією статтею складаються з планового фонду зарплати всіх категорій працівників, зайнятих у розробці програми [22]. Розрахунок зарплати ведеться на підставі даних про трудомісткість, представлених в табл. 6.4.

Таблиця 6.4 – Розрахунок основної заробітної плати

Посада виконавця	Чисельність, чол.	Місячний оклад, грн.	Кількість місяців	Сума заробітної платні, грн
Програміст	1	5950	3,5	20825
Консультант	1	7945	0,25	1986,25
Підсумок	2			22811,25

6.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати працівника, розраховують за формулою (6.2):

$$ЗП_{\text{доп}} = ЗП_{\text{осн}} \cdot 0,1 \quad [22]. \quad (6.2)$$

Підставивши величину основної заробітної плати в дану формулу, отримуємо:

$$ЗП_{\text{доп}} = 22811,25 \cdot 0,1 = 2281,125 \text{ грн.}$$

6.2.3 Відрахування на єдиний соціальний внесок

Відрахування на єдиний соціальний внесок становлять 22% і беруться від основної та додаткової заробітної плати. Вони розраховуються за формулою (6.3):

$$\begin{aligned} ВТ_{\text{соц}} &= (ЗП_{\text{осн}} + ЗП_{\text{доп}}) \cdot 0,22 \quad [22], \\ ВТ_{\text{соц}} &= (22811,25 + 2281,125) \cdot 0,22 = 5520,32 \text{ грн.} \end{aligned} \quad (6.3)$$

6.2.4 Визначення витрат на матеріали

Використовується 3 найменування матеріалів: диск DVD–R – 11 грн., тонер для картриджа – 150 грн. і папір – 88 грн. (1 упаковка).

Витрати на матеріали розраховуються за формулою (6.4):

$$M = \sum_{i=1}^n (\text{Ц}_i \cdot N_i \cdot (1 + K_{\text{т.з.}}) - \text{Ц}_{\text{іо}} \cdot N_{\text{іо}}), \quad (6.4)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Ц_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн.;

N_i – потреба в i -му матеріалі, напівфабрикатах і комплектуючих;

$\text{Ц}_{\text{іо}}$ – ціна зворотних відходів i -го найменування матеріалу, грн.;

$N_{\text{іо}}$ – кількість зворотних відходів i -го найменування;

n – кількість матеріалів, напівфабрикатів та комплектуючих [22].

При цьому

$$\text{Ц}_{\text{іо}} = 0,$$

$$N_{io} = 0,$$

$$K_{т.з.} = 0,05,$$

$$M = (1 + 0,05) \cdot (11 + 150 + 88) = 261,45 \text{ грн.}$$

Разом витрати на матеріали складають 261,45 грн.

6.2.5 Витрати на спеціальне обладнання

У цю статтю входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання [22].

Практично, в даному випадку, в цій статті, враховуються витрати на оплату машинного часу електронно-обчислювальної машини (ЕОМ) для написання і налагодження даної програми, для чого необхідно скласти схему витрат на утримання та експлуатацію устаткування, виходячи з якої визначити вартість однієї машино-години роботи персонального комп'ютера (ПК), після множення якої на машинний час, який пішов на написання і налагодження програми, отримаємо витрати на оплату машинного часу [22].

Амортизаційні відрахування визначаються за формулою (6.5):

$$A = \frac{\Phi_6 \cdot H_a}{100}, \quad (6.5)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн.;

H_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, % [22].

Балансова вартість обчислювальної техніки становить 5000 грн.

Одержуємо:

$$A = 5000 \cdot 0,25 = 1250 \text{ грн.}$$

Статтю «Експлуатація обладнання» розраховують підсумовуванням витрат на електроенергію і допоміжні матеріали на основі формули (6.6):

$$C_e = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зв}} \cdot K_{\text{зм}} \cdot C_e, \quad (6.6)$$

де N_n – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{еф}}$ – річний ефективний фонд часу роботи ЕОМ, машино-годин;

$K_{\text{зв}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$ – коефіцієнт завантаження по потужності;

C_e – ціна одного кВт · рік електроенергії, грн. / (кВт·год.) [22].

Номінальна потужність робочої машини – 0,12 кВт. Річний ефективний фонд часу роботи ЕОМ становить 1800 годин. Середні коефіцієнти завантаження за часом і по потужності рівні відповідно 0,9 і 0,6. Ціна однієї кіловат-години електроенергії становить 2,11 грн.

Отримаємо:

$$C_e = 0,12 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 246,11 \text{ грн}.$$

Зарплати обслуговуючого персоналу розраховуються за формулою (6.7):

$$ЗП_{\text{обсл}} = \frac{\Phi ЗП_{\text{г}} \cdot (1 + K_{\text{отч}}) \cdot t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}}, \quad (6.7)$$

де $\Phi ЗП_{\text{г}}$ – річний фонд заробітної плати (основної і додаткової) обслуговуючих робітників, грн.;

$K_{\text{отч}}$ – коефіцієнт, що враховує відрахування на єдиний соціальний внесок;

$t_{\text{обсл}}$ – час протягом року, необхідний на технічне обслуговування ЕОМ;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, год./рік [22].

Місячна заробітна плата обслуговуючого персоналу становить 3723 грн., а річний фонд заробітної плати відповідно дорівнює 44676 грн. Річний ефективний фонд робочого часу обслуговуючого ПК працівника дорівнює 1750 год/рік. На обслуговування одного ПК витрачається по 1 годині на місяць, що в рік становить 12 годин.

Отримаємо:

$$ЗП_{\text{обсл}} = \frac{44676 \cdot 12 \cdot (1 + 0,22)}{1750} = 373,75 \text{ грн.}$$

Стаття «Поточний ремонт обладнання» приймається рівною 3% від балансової вартості обладнання і становить 150 грн. Стаття «Інші витрати» приймається рівною 5% від суми всіх попередніх статей витрат на утримання і експлуатацію обладнання. Сума всіх попередніх статей дорівнює 2019,86 грн., 5% від суми становлять 100,99 грн. Розраховані статті витрат на утримання та експлуатацію устаткування внесені в табл. 6.5.

Таблиця 6.5 – Кошторис витрат на утримання та експлуатацію обладнання

Найменування статей витрат	Сума, грн.
Амортизація обладнання	1250
Експлуатація обладнання (крім витрат на поточний ремонт)	246,11
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	373,75
Поточний ремонт обладнання	150
Інші витрати	100,99
Підсумок	2120,85

Витрати на оплату машинного часу ЕОМ для написання і налагодження даної програми визначаються за формулою (6.8):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (6.8)$$

де $C_{\text{мо}}$ – витрати на оплату машинного часу, грн.;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино-годин;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино-годин [22].

Експлуатаційні витрати на одну годину машинного часу використовуваної ЕОМ розраховують діленням суми витрат за кошторисом «Витрати на утримання та експлуатацію обладнання (ПК)» на річний ефективний фонд часу роботи ПК. Річний ефективний фонд часу роботи ПК дорівнює 1800 годин. В результаті експлуатаційні витрати на одну годину машинного часу рівні:

$$P_{\text{екс}} = 2120.85/1800 = 1,18 \text{ грн./машино-год.}$$

ЕОМ експлуатується 68 днів в одну зміну, що становить в сумі 544 години. Таким чином, витрати на оплату машинного часу складуть:

$$C_{\text{мо}} = 1,18 \cdot 544 = 641,92 \text{ грн.}$$

6.2.6 Інші прямі витрати

До інших прямих витрат включаються витрати на використовуване при розробці системи комерційне програмне забезпечення: пайове програмне забезпечення, що використовується постійно при роботі ПК (Windows 10 Home) – 3359,2 грн. без податку на додану вартість.

Цільове програмне забезпечення, призначене для даної конкретної задачі, буде отримано на безкоштовних умовах.

Розрахунок проводиться за формулою (6.9):

$$S_{най.ПО} = \frac{Ц_{ПОWindows} \cdot T_{KTC}}{\Phi_{эф.КТС} \cdot T_{с.ПО}}, \quad (6.9)$$

$$S_{цел.ПО} = 0 \text{ грн.}$$

де $S_{най.ПО}$ – витрати на пайове програмне забезпечення при розробці програмного забезпечення з розрахунків, грн;

$S_{цел.ПО}$ – витрати на цільове програмне забезпечення, що купується виключно для розробки програмного забезпечення з розрахунків, грн;

$Ц_{ПОWindows}$ – ціна за Windows (без податку на додану вартість), грн;

T_{KTC} – машинний час комплексу технічних засобів, необхідний користувачеві для розробки системи, машино-год/рік;

$\Phi_{эф.КТС}$ – річний ефективний фонд часу роботи комплексу технічних засобів, машино-год/рік;

$T_{с.ПО}$ – термін служби пайового програмного забезпечення, років [22].

Розрахуємо:

$$S_{най.ПО} = \frac{3359,2 \cdot 544}{1800 \cdot 5} = 203,04 \text{ грн.},$$

$$S_{\Sigma} = 203,04 + 0 = 203,04 \text{ грн.}$$

6.2.7 Розрахунок накладних витрат

Результати визначення витрат на розробку програми у вигляді калькуляції кошторисної вартості робіт наведені в табл. 6.6.

Таблиця 6.6 – Калькуляція кошторисної вартості робіт з розробки програми

Найменування статей витрат	Сума, грн.	Питома вага до підсумку, %
Основна заробітна плата	22811,25	59,15
Додаткова заробітна плата	2281,125	5,91
Відрахування на єдиний соціальний внесок	5520,32	14,31
Матеріали та комплектуючі	261,45	0,68
Витрати на спеціальне обладнання	641,92	1,66
Інші прямі витрати	203,04	0,52
Накладні витрати	6843,38	17,75
Разом	38562,49	100

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т. д.), утримання та експлуатацію будівель, накладні витрати включаються у вартість розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників [22]. В даному випадку накладні витрати складають 30% до основної заробітної плати розробників, що становить 6843,38 грн.

6.3 Оцінка економічної ефективності

Програмний продукт дозволяє скоротити час і трудомісткість процесу тестування програмно-апаратних засобів мінікомп'ютерних систем [22].

Економічна ефективність досягається за рахунок скорочення часу аналізу з 500 до 200 годин на рік, що в сумі призводить до економії

заробітної плати тестувальника, що складає 90 грн на годину. Річна економія по заробітній платі з урахуванням начислень становить:

$$E_{\text{зп}} = (500 - 200) \cdot 90 \cdot (1 + 0,22) = 32940,00 \text{ грн.}$$

Річну економію від впровадження програми, що дозволяє знизити витрати машинного часу ЕОМ для рішення певного завдання в порівнянні із програмою, що застосовувалася раніше, визначають за формулою (6.10):

$$E = (T_{M_1} - T_{M_2}) \cdot B_{\text{екс.КТЗ}} + E_{\text{н}}^a \cdot \frac{\Phi_{\text{б.КТЗ}} \cdot (T_{M_1} - T_{M_2})}{\Phi_{\text{еф.КТЗ}}} - \frac{S_{\text{рп}}}{T_{\text{с}}} + E_{\text{зп}}, \quad (6.10)$$

де T_{M_1}, T_{M_2} – машинний час, необхідний для рішення поставлених завдань, відповідно в старому та у новому варіантах, машино-год./рік;

$B_{\text{екс}}$ – експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн/ машино-год.;

$E_{\text{н}}^a$ – нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{\text{б.КТЗ}}$ – балансова вартість ЕОМ, грн.;

$\Phi_{\text{еф.КТЗ}}$ – річний ефективний фонд часу роботи ЕОМ, машино-год.;

$S_{\text{рп}}$ – сумарні витрати на розробку програми, грн.;

$T_{\text{с}}$ – термін служби впроваджуваної програми до її морального зношування, років;

$E_{\text{зп}}$ – економія на заробітній платі [22].

Розрахуємо:

$$E = (500 - 200) \cdot 1,18 + 0,33 \cdot 5000 \cdot (500 - 200) / 1800 - 38562,49 / 3 + 32940,00 = 20714,84 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень E і термін їх окупності $T_{ок}$ розраховуються за формулами (6.11) і (6.12):

$$E\phi = \frac{E}{S_{pp}}, \quad (6.11)$$

$$T_{ок} = \frac{S_{pp}}{E} [22]. \quad (6.12)$$

Таким чином:

$$E = 20714,84 / 38562,49 = 0,54,$$

$$T_{ок} = 38562,49 / 20714,84 = 1,86.$$

Термін окупності програми становить 1,86 року, що менше нормативного терміну окупності. Річна економія при використанні програми становить 20714,84 грн. Таким чином, впровадження програмного продукту є економічно доцільним.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

В роботі проводиться дослідження та програмна реалізація методів рекомендації фільмів.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники:

- невідповідність ергономічних характеристик устаткування нормованим величинам внаслідок нераціональної організації робочого місця;

- негативний вплив випромінювань відео дисплейних терміналів на органи зору людини має основною причиною перенапруження м'язового апарату ока, що пов'язане з недостатньою контрастністю та мерехтінням зображення, відблесками та відбитим світлом від моніторів, крім того, втома очей настає при частій зміні огляду більш освітленого монітора та менш освітленої клавіатури, що призводить до «астенопії» (відчуття дискомфорту та біль в очах);

- негативний вплив при роботі з відео дисплейними терміналами на функціональний стан людини який обумовлений електростатичними полями, які створюються моніторами з електронно-променевою трубкою, що може призвести до зниження імунітету, захворювань периферійної нервової системи або бронхо-легеневих та шкіряних захворювань;

- можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричному ланцюзі, тому у приміщенні можуть виникнути аварійні ситуації: замикання, загоряння проводки, поразення електрострумом, що може привести до електротравм або летальних наслідків;

- статична електрика;

- невідповідність нормам параметрів мікроклімату;

- незадовільні умови освітлення робочого місця, в наслідок не рівномірного розподілу освітлювальних приладів;

- небезпеки, пов'язані з підвищеним рівнем шуму, який створюється устаткуванням та людьми, що знаходяться у приміщенні, і які призводить до погіршення слуху;
- можливість загорянь внаслідок порушень правил пожежної безпеки, що може призвести до пожеж;
- неправильні дії персоналу під час надзвичайних ситуацій внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків [23].

7.2 Заходи по забезпеченню безпеки

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

У розділі 4 Державних санітарних правил і норм З.З.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» передбачені гігієнічні вимоги до організації устаткування робочих місць візуальних дисплейних терміналів (ВДТ) ЕОМ і персональних ЕОМ, конструкція робочого стола повинна відповідати сучасним вимогам ергономіки й забезпечувати оптимальне розміщення на робочій поверхні використовуваного устаткування (дисплея, клавіатури, принтера) і документів, висота робочої поверхні робочого стола із ВДТ повинні регулюватися в межах 680...800 мм, а ширина й глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800..1000 мм), приймаємо робочу поверхню з розмірами 1400x800, щоб уникнути відблисків на екрані вікон доцільно розмістити робоче місце торцем до вікна не ближче чим на 1 м, площа дозволяє нам розмістити додаткові меблі для зберігання документів, магнітних дисків, полками, стелажми, тумбами й т.п. [23].

Велике значення надається ергономічним та естетичним вимогам – цього домоглися за допомогою колірного рішення приміщення й устаткування з урахуванням вимог ГОСТ 17.2.032-78 «Робоче місце при виконанні робіт сидячи» – у приміщеннях стіни пофарбовані фарбами пастельних тонів, під час організації робочого простору необхідно враховувати індивідуальні антропометричні параметри користувача з відповідними допусками на можливі зміни робочих поз та потребу у переміщеннях [23].

Заходи мінімізації негативного впливу випромінювань від відео дисплейних терміналів на органи зору, передбачено такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці:

- для розробників програм із застосуванням ЕОМ слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;

- для операторів із застосування ЕОМ слід призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;

- для операторів комп'ютерного набору слід призначати регламентовані перерви для відпочинку тривалістю 10 хвилин після кожною години роботи за ВДТ [24].

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 години.

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин [24].

Відзначимо наступне:

– з метою зменшення негативного впливу монотонності є доцільним застосовувати чергування операцій усвідомленого тексту і числових даних (зміна змісту роботи);

– для зниження нервово-емоційного напруження, втомлення зорового аналізатору, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільні деякі перерви використовувати для виконання комплексу вправ;

– в окремих випадках – при хронічних скаргах працюючих з ВДТ ЕОМ і ПЕОМ на зорове втомлення, незважаючи на дотримання санітарно-гігієнічних вимог до режимів праці і відпочинку, а також застосування засобів локального захисту очей – допускаються індивідуальних підхід до обмеження часу робіт з ВДТ, зміни характеру праці, чергування з іншими видами діяльності, не пов'язаними з ВДТ;

– активний відпочинок має полягати у виконанні комплексу гімнастичних вправ, спрямованих на зняття нервового напруження, м'язове розслаблення, відновлення функцій фізіологічних систем, що порушуються протягом трудового процесу, зняття втоми очей, поліпшення мозкового кровообігу і працездатності [24].

Згідно правил улаштування електроустановок (ПУЕ) приміщення виробництва відноситься до категорії «Приміщення без підвищеної небезпеки» [24].

Апаратура, що перебуває під небезпечною для життя напругою, припускає забезпечення електробезпечності, всі металеві частини повинні бути заземлені на третій провід спеціальних триконтактних розеток, струмопровідні кабелі повинні бути розташовані під дерев'яною підлогою, зверху покритою лінолеумом, або у вигляді схованої проводки в стінах, таким чином, контакт із кабелем виключається, над кожною розеткою повинен бути попереджувальний надпис [24].

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення, у процесі експлуатації періодично

проводиться перевірка якості ізоляції, шляхом виміру її опору, відповідно до ПУЕ в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5МОм [25].

Відповідно до ПУЕ захисне заземлення 4 Ом застосовується у всіх електроустановках змінного струму під напругою 220 В і вище й постійного струму 440 В і вище, у приміщеннях з підвищеною небезпекою заземлення використовується в електроустановках змінного струму напругою 42 В та постійного струму 110 В, захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок відповідно до ГОСТ 12.1.030-81 «Електробезпека. Захисне заземлення. Занулення» [25].

Відзначимо наступне:

- вирівнювання потенціалів досягаються шляхом устрою контурних заземлювачів, у приміщеннях вирівнювання потенціалів здійснюється з'єднанням металевих стійок ЕОМ, корпусів апаратів світильників і іншого устаткування з усіма доступними для дотику металевими конструкціями будинку, спорудження, а також з контуром шиною захисного заземлення;

- електричний поділ мережі припускає розділ мережі на окремі електрично непов'язані між собою ділянки за допомогою розділових трансформаторів, підвищення електробезпечності відбувається за рахунок зменшення ємкості мережі й підвищення опору ізоляції фаз мережі щодо землі;

- подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, що ізолює людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою, при застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їхніх корпусів;

– накопичена статична електрика, зокрема, на екрані монітора притягує пил, бруд та інші частини присутні в повітрі, причому електризується не тільки екран, а і повітря на робочому місці, а також одяг користувача, особливо якщо він з синтетичного та шерстяного матеріалу [25].

В якості заходів захисту запропоновано:

- кілька разів на протязі робочого дня мити руки і обличчя водою, а після закінчення роботи вимити руки й лице з милом;
- щоденно протирати екран монітора, клавіатуру, пристрій «миша», а якщо є прикраний фільтр, то і його антистатичною серветкою;
- щоденно в приміщенні з ПК проводити вологе прибирання;
- установити нейтралізатори статичної електрики;
- підтримувати у приміщенні вологість повітря зазначену в нормативних документах;
- користувачу ПК бажано носити одяг з природних (льняних) волокон, в т.ч. і шкарпетки [26].

7.3 Заходи по забезпеченню промислової санітарії та гігієни праці

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно Державних санітарних норм 3.3.6.042-99 санітарні норми мікроклімату виробничих приміщень – 20...23°C, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 22...25°C, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек.

Для забезпечення вищевказаних параметрів передбачене застосування вентиляції – природної (аерації й провітрювання), механічної (загальнообмінної, місцевої приточної і витяжної), кондиціонування повітря.

У приміщенні забезпечений приплив свіжого повітря, кількість якого визначається техніко-економічним розрахунком і вибором схеми системи

вентиляції. Мінімальна витрата повітря визначається з розрахунку 50...60 м³/ч на одну людину, але не менш дворазового повітрообміну в годину.

При невеликому забрудненні зовнішнього повітря кондиціонування приміщень здійснюється із перемінними витратами зовнішнього повітря й рециркуляційного.

Всі норми по освітленості беруться згідно Державних будівельних норм В.2.5-28-2006 «Природне та штучне освітлення». Штучне освітлення в приміщеннях здійснюється системою загального рівномірного освітлення. При цьому штучне освітлення застосовують не тільки в темний, але й у світлий час доби.

Найбільш прийнятними для приміщень є люмінесцентні лампи ЛБ (білого світла) потужністю 20, 40 або 80 Вт, які встановлюються в світильники типу ЛСП. Допускається застосування ламп накаливання у світильниках місцевого освітлення.

Загальне освітлення варто виконувати у виді суцільних чи переривчастих ліній світильників, розташованих збоку від робочих місць.

Розрахуємо систему загального рівномірного штучного освітлення в адміністративному приміщенні. Приймаємо розміри приміщення довжина $A = 11$ м, ширина $B = 7$ м, висота $H = 5$ м. Висота робочої поверхні складає $h = 0,7$ м.

Коефіцієнти відбиття поверхонь приміщення: колір стелі, стін - сірий, незначне пиловиділення, $\rho_c = 50\%$; $\rho_{ст} = 30\%$; $\rho_n = 10\%$,

Розряд зорової роботи – III б.

Згідно з ДБН В.2.5-28-2006 «Природне і штучне освітлення. Норми проєктування» рівень нормованого загального освітлення в приміщенні нормуються як $E_n = 400$ Лк.

Обираємо світильник типу ЛПП, призначений для приміщень висотою до 6 м. $IP = 65$ – ступінь захисту світильника, $[L/h] = 1,4$ – числове значення коефіцієнта світильника.

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,6$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Визначається кількість рядів світильників у приміщенні за формулою (7.1):

$$N_p = \frac{B}{(H - h_p) \cdot (L / h)} = 1,16 \approx 2. \quad (7.1)$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (7.2):

$$L_{\max} = \frac{B}{N_p}, \quad (7.2)$$

де B – ширина приміщення:

N_p – кількість рядів.

Розрахуємо:

$$L_{\max} = \frac{7}{2} = 3,5 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (7.3):

$$h = \frac{L_{\max}}{[L / h]}, \quad (7.3)$$

$$h = \frac{3,5}{1,4} = 2,5 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі за формулою (7.4):

$$h_3 = H - h_p - h, \quad (7.4)$$

де H – висота приміщення;

h_p – висота робочої поверхні;

h – висота підвісу світильника над робочою поверхнею.

Розрахуємо:

$$h_3 = 5 - 0,7 - 2,5 = 1,8 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (7.5):

$$i = \frac{AB}{h(A+B)}, \quad (7.5)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м.

Розрахуємо:

$$i = \frac{11 \cdot 7}{2,5 \cdot (11 + 7)} = 1,71.$$

Значення коефіцієнта використання світлового потоку η вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення (ρ_C , ρ_{CT} , ρ_{II}) та індексу приміщення. В нашому випадку $\eta = 54\%$.

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні за формулою (7.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_3 \cdot z}{\eta}, \quad (7.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм.

Розрахуємо:

$$\Phi_{\Sigma} = \frac{400 \cdot 11 \cdot 7 \cdot 1,6 \cdot 1,1}{0,54} = 100385 \text{ лм.}$$

Визначаємо умовну загальну кількість світильників у приміщенні за формулою (7.7):

$$N^* = A \cdot \frac{B}{L_{\max}^2}, \quad (7.7)$$

$$N^* = \frac{11 \cdot 7}{3,5 \cdot 3,5} = 6,3 \approx 8 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла за формулою (7.8):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм,} \quad (7.8)$$

де $N_{\text{л}}$ – загальна кількість ламп у приміщенні, *шт*;

При цьому за формулою (7.9):

$$N_{\text{л}} = N^* \cdot n, \quad (7.9)$$

де n – кількість ламп у світильнику (в ЛПП світильнику знаходиться 2 лампи).

Розрахуємо:

$$N_{\text{л}} = 8 \cdot 2 = 16 \text{ шт.}$$

Тоді

$$\Phi_{\text{л}} = \frac{100385}{16} = 6274 \text{ лм.}$$

Обираємо тип стандартної лампи з найближчим значенням фактичного світлового потоку лампи $\Phi_{\text{л}}$ і знайти коефіцієнт m (співвідношення між розрахунковим світловим потоком лампи $\Phi_{\text{л}}^*$ та фактичним світловим потоком вибраної стандартної лампи $\Phi_{\text{л}}$).

Обираємо лампу ЛБ із $P_{\text{л}}=40$ Вт, $\Phi_{\text{л}} = 5800$ лм, за формулою (7.10):

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}}, \quad (7.10)$$

$$m = \frac{6274}{5800} = 1,081.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (7.11):

$$N = N^* \cdot m, \quad (7.11)$$

$$N = 8 \cdot 1.081 \approx 8 \text{ шт.}$$

Фактична кількість ламп у приміщенні розраховується за формулою (7.12):

$$N_{\text{фл}} = N_{\text{ф}} \cdot n, \quad (7.12)$$

$$N_{\text{фл}} = 8 \cdot 2 = 16 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп, за формулою (7.13):

$$E_p = \frac{\Phi_{\text{л}} N_{\text{фл}} \eta}{S k_z z}, \quad (7.13)$$

$$E_p = \frac{5800 \cdot 16 \cdot 0,54}{11 \cdot 7 \cdot 1,6 \cdot 1,1} = 370 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_{\text{н}}$, лк.

Таким чином, на основі розрахунків для освітлення приміщення площею 77 м² встановлено 8 світильників, розташованих у 2 рядах по 4 світильники у кожному ряді, що дозволяє отримати освітленість на робочому місці 400 лк відповідно до Державних будівельних норм В.2.5-28-2006 «Природне і штучне освітлення. Норми проектування».

Згідно з Державними санітарними нормами 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливим завданням. Зниження шуму в джерелі виникнення забезпечується застосуванням пружних прокладок між основою машини, приладу й опорною поверхнею. Як прокладки використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, – прокладки з м'якої гуми, повсті, товщиною 6 – 8 мм.

7.4 Заходи по забезпеченню пожежної безпеки

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог Нормативно-

правових актів з пожежної безпеки А.01.001-2014 «Правила пожежної безпеки в Україні».

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

- згідно ГОСТ 27331-87 «Пожежна техніка. Класифікація пожеж» у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е;

- відповідно до вимог ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою», воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали [26].

Було прийнято наступні рішення:

- оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог Державних будівельних норм В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» воно має II ступінь вогнестійкості;

- у разі виникнення пожежі у приміщенні обладнаному ПК з ВДТ для евакуації персоналу відповідно до вимог Державних будівельних норм В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – входні двері, згідно п. 2.29 Сантіарних норм і правил 2.09.02-85 «Виробничі будівлі», відстань від найбільш віддаленого робочого місця до найближчого евакуаційного виходу не обмежується;

- обладнання, силові та освітленні мережі приміщення, обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог Нормативно-правових актів з пожежної безпеки 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок», та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежонебезпечної зони П-Па до якої належить приміщення;

– для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації, вони можуть самостійно пускати в хід установки пожежогасіння, коли пожежа не досягла великих розмірів, системи автоматичної пожежної сигналізації складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій), згідно вимог Державних будівельних норм В.2.5-56:2014 «Системи протипожежного захисту», в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика;

– оскільки приміщення (дослідницької лабораторії, конструкторського бюро, тощо) що обладнане ПК з ВДТ має площу 77 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 4 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг і більше, на 20 м² площі приміщення), відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м [26].

7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях

Укриття населення у захисних спорудах цивільного захисту недотримання якого може призвести до механічних травм, зараження хімічними викидами або летального наслідку.

Укриття населення в захисних спорудах – це комплекс заходів із завчасним будівництвом захисних споруд, а також пристосуванням наявних приміщень для захисту населення та підтримання їх у готовності до використання [27].

Виділимо наступне:

– укриттю в захисних спорудах у надзвичайних ситуаціях підлягає все населення України, фонд захисних споруд створюється шляхом обстеження й обліку підземних та наземних будівель і споруд, що відповідають вимогам захисту населення; дообладнання з урахуванням реальної обстановки підвалів, погребів та інших заглиблених приміщань; обстеження і взяття на облік підземних і наземних будівель та споруд гірничих виробок і природних порожнин, що відповідають вимогам захисту; у разі необхідності переобладнання цих приміщень; будівництво заглиблених споруд пристосованих для захисту, що окремо розташовані від об'єктів виробничого призначення; масового будівництва в період загрози надзвичайних ситуацій найпростіших сховищ та укриттів; будівництво окремих сховищ та протирадіаційних укриттів;

– центральний орган виконавчої влади, до компетенції якого належать питання захисту населення і територій від надзвичайних ситуацій визначає перелік сховищ, протирадіаційних укриттів та інших захисних споруд, які необхідно будувати, що затверджується Кабінетом Міністрів України;

– потреби в захисних спорудах визначають, виходячи з необхідності укриття всіх працюючих за місцем роботи і проживання, усього непрацюючого населення за місцем проживання;

– укриття населення в захисних спорудах є надійним способом захисту від уражаючих факторів ядерної, хімічної, бактеріологічної, звичайної зброї, у разі аварій і деяких стихійних лих (ураганів, снігових заносів) [28].

Загалом можна виділити:

– захисні споруди за своїм призначенням і захисними властивостями поділяються на сховища, протирадіаційні укриття (ПРУ) і найпростіші укриття – щілини;

– сховища і протирадіаційні укриття будують завчасно, вони мають подвійне призначення: для потреб об'єктів народного господарства (навчальні класи, для спортивних секцій та ін.) і укриття населення;

– сховища це інженерні споруди, які забезпечують надійний захист людей від усіх уражаючих факторів ядерного вибуху, отруйних і сильнодіючих ядучих речовин, бактеріальних засобів і уражаючих факторів звичайної зброї, обвалів і уламків зруйнованих будівель і споруд;

– класифікуються вони за захисними властивостями, місткістю, місцем розміщення, забезпеченням фільтровентиляційним обладнанням і часом побудови;

– за захисними властивостями (від дії вибухової хвилі) сховища поділяються на п'ять класів;

– за місткістю сховища поділяються на: малі – до 150 осіб, середні – від 160 до 450, великі – понад 450 осіб;

– за місцем розташування – на вбудовані, які розміщені у підвальних приміщеннях будівель, та окремо побудовані поза будівлями;

– за забезпеченням фільтровентиляційним обладнанням – промислового виготовлення і спрощене, виготовлене з підручних матеріалів;

– за часом побудови на: побудовані завчасно і швидко споруджені;

– ПРУ – це захисна споруда, яка забезпечує захист у ній людей від радіоактивних речовин і опромінення в зонах радіоактивного забруднення місцевості, отруйних і сильнодіючих ядучих речовин, біологічних засобів у краплинно-рідинному вигляді та світлового випромінювання ядерного вибуху, наслідків урагану;

– захисні властивості протирадіаційних укриттів оцінюються коефіцієнтом захисту, який показує, у скільки разів доза радіації на відкритій місцевості на висоті 1 м більша від дози радіації в укритті, тобто коефіцієнт захисту показує, у скільки разів ПРУ послаблює дію радіації, а відповідно і дозу опромінення людей;

– протирадіаційні укриття можуть обладнуватись насамперед у підвальних поверххах будинків і споруд, підвали в дерев'яних одноповерхових будинках ослаблюють дозу радіації в 7 разів, а в житлових одноповерхових кам'яних (цегляних) будинках – у 40, у двоповерхових – у 100, середня частина підвалу кількоповерхового кам'яного будинку – у 800 – 1000 разів, при невисоких рівнях радіації, а також для захисту від бактеріальних засобів, парів отруйних і сильнодіючих ядухих речовин можна використовувати кам'яні (цегляні) або дерев'яні будівлі [23], [29].

Таким чином, відповідно до ПУЕ захисне заземлення 4 Ом застосовується у всіх електроустановках змінного струму під напругою 220В і вище.

На основі розрахунків для освітлення приміщення площею 77 м² встановлено 8 світильників, розташованих у 2 рядах по 4 світильники у кожному ряді, що дозволяє отримати освітленість на робочому місці 400лк відповідно до Державних будівельних норм В.2.5-28-2006 «Природне і штучне освітлення. Норми проектування».

Згідно з Державними санітарними нормами 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» зниження шуму в джерелі виникнення забезпечується застосуванням пружних прокладок між основою машини, приладу й опорною поверхнею товщиною 6-8 мм.

Відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 4 штуки.

ВИСНОВКИ

У роботі розглянуто проблему рекомендування фільмів. Виконано огляд типів і принципів рекомендаційних систем. Проаналізовано сучасні програмні системи рекомендацій фільмів, які використовуються в сервісах IMDb, Netflix, Moviestape, виділено їх основні функціональні можливості. Виконано аналіз методів рекомендування фільмів, принципів організації рекурентних нейронних мереж, зокрема довгої короткочасної пам'яті. Проаналізовано метод word2vec, який дозволяє виконувати кодування слів у вигляді вектору.

Обрано мову програмування C# для створення Windows-застосунку з графічним інтерфейсом користувача та мову програмування Python для створення засобів реалізації запропонованого методу. Визначено функціональні вимоги до розроблюваного програмного забезпечення. Представлено діаграму прецедентів для роботи неавторизованого, авторизованого користувачів та адміністратора з системою. Обрано MySQL в якості системи керування базами даних для проєктування та реалізації бази даних. Виконано проєктування бази даних з представленням відповідної схеми бази даних.

Визначено структуру програмної системи, що складається з підсистеми для організації моделі предметної області, підсистеми графічного інтерфейса користувача, підсистеми рекомендування фільмів та бази даних. Розроблено програмне забезпечення.

Наукова новизна роботи полягає в запропонованому методі рекомендування фільмів, який характеризується виконанням рекомендування фільмів на основі використання довгої короткочасної пам'яті та використання в якості її вхідних ознак закодованих векторних представлень фільмів за допомогою використання методу word2vec, що дозволяє виконувати рекомендування фільмів з врахуванням послідовності уподобань користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Рекомендательные системы сегодня – необходимость для бизнеса [Электрон. ресурс]. – Режим доступа : <https://blog.heyml.com/рекомендательные-системы-сегодня-необходимость-для-бизнеса-41c1c9109988>.
2. Анатомия рекомендательных систем. Часть первая [Электрон. ресурс]. – Режим доступа : <https://habr.com/ru/company/lanit/blog/420499/>.
3. Системы рекомендаций [Электрон. ресурс] : введение в гибридные системы. – Режим доступа : <https://habr.com/ru/post/58309/>.
4. Филиппова, Е. Системы рекомендаций: задачи, подходы, алгоритмы [Электрон. ресурс] / Е. Филиппова. – Режим доступа : <http://datareview.info/article/sistemyi-rekomendatsiy-zadachi-podhodyi-algoritmyi/>.
5. IMDb [Electronic resource] : Scores, Reviews, and Where to Watch the Best Movies & TV Shows. – Access mode : <https://www.imdb.com/>.
6. IMDb Movies & TV [Электрон. ресурс]: Все что вы хотели знать о кинематографе. – Режим доступа : <https://apps4.life/imdb-movies-tv-vse-chto-vi-hoteli-znat-o-kinematografe/>.
7. Moviestape – кіно онлайн українською мовою [Електрон. ресурс]. – Режим доступу : <http://moviestape.net/>.
8. Netflix Ukraine – Watch TV Shows Online, Watch Movies Online [Electronic resource]. – Access mode : <https://www.netflix.com/ua/>.
9. Все о Netflix. Что это за приложение и стоит ли его покупать? [Электрон. ресурс]. – Режим доступа : <https://deep-review.com/articles/what-is-netflix/>.
10. Melanson, D. Netflix app for Windows 8 now available for download [Electronic resource] / D. Melanson. – Access mode : <https://www.engadget.com/2012-10-12-netflix-app-for-windows-8-now-available-for-download.html>.

11. Isinkayea, F.O. Recommendation systems [Text] : Principles, methods and evaluation / F.O.Isinkayea, Y.O.Folajimib, B.A.Ojokohc // Egyptian Informatics Journal. – 2015. – Volume 16, Issue 3. – Pp. 261-273.
12. Разнообразие нейронных сетей. Часть вторая. Продвинутое конфигурации [Электрон. ресурс]. – Режим доступа : <https://tproger.ru/translations/neural-network-zoo-2/>.
13. Phi, M. Illustrated Guide to LSTM's and GRU's: A step by step explanation [Electronic resource] / M. Phi. – Access mode : <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
14. Немного про word2vec [Электрон. ресурс] : полезная теория. – Режим доступа : <http://nlpx.net/archives/179>.
15. Word2Vec [Электрон. ресурс] : как работать с векторными представлениями слов. – Режим доступа : <https://neurohive.io/ru/osnovy-data-science/word2vec-vektornye-predstavleniya-slov-dlja-mashinnogo-obuchenija/>.
16. Cosine Similarity – an overview [Electronic resource]. – Access mode : <https://www.sciencedirect.com/topics/computer-science/cosine-similarity>.
17. What is TF-IDF? [Electronic resource]. – Access mode : <https://monkeylearn.com/blog/what-is-tf-idf/>.
18. .NET Core(C#) vs Python [Electronic resource] : What Should You Choose? – Access mode : <https://medium.com/swlh/net-core-c-vs-python-what-should-you-choose-ee5aec46f763>.
19. Язык программирования C# [Электрон. ресурс] : история, специфика, место на рынке. – Режим доступа : <https://geekbrains.ru/posts/yazyk-programmirovaniya-c-sharp-istoriya-spezifika-mesto-na-rynke>.
20. MySQL Workbench Tutorial & MySQL Introduction [Electronic resource]. – Access mode : <https://www.guru99.com/introduction-to-mysql-workbench.html>.

21. 8 Advantages of Using MySQL [Electronic resource]. – Access mode : <https://devops.com/8-advantages-using-mysql/>.
22. Остапенко, В.В. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проєктування» і 7.091501 «Комп'ютерні та інтелектуальні системи і мережі» всіх форм навчання [Текст] / В.В. Остапенко, Е.М. Касьян, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2015. – 22 с.
23. Катренко, Л. А. Охорона праці. Курс лекцій. Практикум [Текст] : навчальний посібник / Л. А. Катренко, Ю. В. Кіт, І. П. Піскун. – 2-ге вид., стер. – Суми : Університетська книга, 2007. – 203 с.
24. Правила улаштування електроустановок [Текст] : Нормативне виробничо-практичне видання : вид. 5-те, перероблене й доповнене : затв. М-вом енергетики та вугільної промисловості України 20.06.14 : введення в дію з 20.11.14. – Х. : Міненерговугілля України, 2014. – 793 с.
25. Жидецький, В. Ц. Основи охорони праці [Текст] : підручник / В. Ц. Жидецький. – 5-те вид., доп. – К. : Знання, 2014. – 373 с.
26. Основи охорони праці [Текст] : підручник / В. В. Березуцький, Т. С. Бондаренко, Г. Г. Валенко та ін. ; за ред. проф. В. В. Березуцького. – Х. : Факт, 2005. – 327 с.
27. Кодекс цивільного захисту України [Текст] : [прийнятий ВР України 02.10.2012 р. № 5403-VI] : офіц. текст : станом на 06.12.2012 р. / ВР України. – К. : Алерта, 2012. – 120 с.
28. Шоботов, В. М. Цивільна оборона [Текст] : навчальний посібник / В. М. Шоботов. – Вид. 2-ге, перероб. – К. : Центр навчальної літератури, 2006. – 438 с.
29. Стеблюк, М. І. Цивільна оборона та цивільний захист [Текст] : навч. посіб. для вузів / М. І. Стеблюк. – К. : Знання, 2013. – 487 с.

ДОДАТОК А
Технічне завдання

Вступ

У дипломній кваліфікаційній роботі магістра необхідно виконати дослідження проблеми рекомендування фільмів, для чого повинен бути виконаний аналіз проблеми на даний момент, методів, які використовуються для рекомендування, а на основі виконаного аналізу повинен бути запропонований власний метод і виконана його програмна реалізація.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу магістра за темою «Дослідження та програмна реалізація методів рекомендування фільмів», затверджене наказом № 301 від 3 листопада 2020 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програмне забезпечення рекомендування фільмів має використовуватись для отримання широкої інформації про випущені в світі фільми, створення власного переліку для подальшого перегляду та отримання рекомендацій для подальшого перегляду.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати виконання наступних функцій:

- додавати та редагувати дані фільмів;
- вносити в систему та редагувати інформацію про акторів, режисерів, продюсерів, жанри;
- переглядати інформацію про фільми;

- виконувати пошук фільмів;
- додавати власні відгуки про фільми;
- переглядати всі відгуки про фільм;
- визначати середню оцінку фільму;
- додавати фільми до переліку для подальшого перегляду;
- переглядати та редагувати перелік для подальшого перегляду;
- оцінювати фільми;
- переглядати рейтинг найкращих фільмів за окремими жанрами або загалом;
- надавати рекомендації для перегляду наступних фільмів;
- виконувати авторизацію облікового запису;
- виконувати реєстрацію облікового запису.

Вхідними даними є дані фільмів, параметри пошуку фільмів, дані про оцінки фільмів та їх відгуки.

Вихідними даними є дані фільмів, рекомендації для перегляду фільмів.

A.3.2 Умови експлуатації

Для роботи програмного забезпечення необхідно на серверній частині інсталиювати:

- інтерпретатор мови програмування Python версії 3;
- середовище Microsoft Visual Studio 2019;
- систему керування базами даних MySQL;
- пакет tensorflow.

A.3.3 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення необхідні наступні технічні засоби:

- процесор з 4 ядрами, при цьому частота кожного має бути не менше 3 Гц;
- оперативна пам'ять – не менше 4 Гб;
- жорсткий диск – вільний простір не менше 2 Гб.

A.3.4 Вимоги до транспортування та збереження

Вимоги до транспортування і зберігання аналогічні вимогам, пропонованим до носія, на якому зберігається програма. Додаткових вимог не висувається.

A.3.5 Вимоги до програмної документації

Програма повинна поставлятися разом з наступними документами:

- текстом програми;
- описом експлуатації та тестування програми.

A.3.6 Вимоги до маркування та упакування

Програма може бути записана на оптичному, гнучкому або жорсткому магнітному диску. У випадку розповсюдження програмного забезпечення на відповідному носії має бути вказано маркування «Дослідження та програмна реалізація методів рекомендації фільмів».

A.4 Стадії та етапи розробки

Виконання дипломної кваліфікаційної роботи магістра повинно бути реалізовано на основі наступних етапів:

- розроблення методу рекомендації фільмів;
- аналіз вимог до програмного забезпечення;

- проєктування програмного забезпечення;
- реалізація програмного забезпечення;
- тестування і відлагодження програмного забезпечення;
- оформлення документації.

Програмний продукт повинен бути виконаний в терміни, визначені календарним планом, наведеним у завданні на роботу.

A.5 Порядок контролю та приймання

Дипломна кваліфікаційна робота магістра повинна бути виконана у відповідності з календарним планом, зазначеним у завданні на роботу.

Дипломна кваліфікаційна робота магістра має бути прийнята консультантами відповідних розділів, після чого затверджена завідуючим кафедрою, а на 13-й тиждень захищена на засіданні екзаменаційної комісії за відповідним графіком.

ДОДАТОК Б
Текст програми

Б.1 Текст файла Actor.cs

```
using System.IO;

namespace FilmRecomModel
{
    public class Actor
    {
        private int id;
        private string surname;
        private string role;
        private string photo;

        public Actor()
        {
            this.Role = "";
            this.Surname = "";
            this.Photo = "";
            this.Id = 0;
        }

        public string Role
        {
            get => role;
            set => role = value;
        }

        public int Id
        {
            get => id;
            set => id = value;
        }

        public string Surname
        {
            get => surname;
            set => surname = value;
        }

        public string Photo
        {
            get => photo;
            set => photo = value;
        }
    }
}
```

```

    }
}

```

Б.2 Текст файла Comment.cs

```

using System.IO;

namespace FilmRecomModel
{
    public class Comment
    {
        private int id;
        private string ctext;
        private Film cfilm;
        private User author;
        private DateTime dtime;

        public Comment()
        {
            this.Id = 0;
            this.Ctext = "";
            this.Author = new User();
            this.Cfilm = new Film();
            this.Dtime = new DateTime();
        }

        public string Ctext
        {
            get => ctext;
            set => ctext = value;
        }

        public Film CFilm
        {
            get => cfilm;
            set => cfilm = value;
        }

        public DateTime Dtime
        {
            get => surname;
            set => surname = value;
        }

        public int Id
        {

```

```

        get => id;
        set => id = value;
    }
    public User Author
    {
        get => author;
        set => author = value;
    }
}
}

```

Б.3 Текст файла Film.cs

```

using System.Collections.Generic;
using System.IO;

namespace FilmRecomModel
{
    public class Film
    {
        private int id;
        private string title;
        private double duration;
        private string fimage;
        private string motto;
        private string dataprem;
        private string[] description = new string[10];
        private string agelevel;
        private double fees;
        private double budget;
        private double avscore;
        private string devcount;
        private List<Actor> actors = new List<Actor>();
        private List<Genre> genres = new List<Genre>();
        private List<Comment> comments = new List<Comment>();
        private List<Rate> scores = new List<Rate>();
        private List<Person> producers = new List<Person>();
        private List<Person> directors = new List<Person>();

        public Film()
        {

```

```

        this.Id = 0;
        this.Title = "";
        this.Dataprem = "01.10.2020";
        this.Budget = 0.0;
        this.Devcount = "";
        this.Motto = "";
        this.Description = [""];
        this.Fees = 0.0;
        this.Duration = 0.0;
        this.Agelevel = "";
        this.Fimage = "";
        this.Avscore = "";
    }
    public List<Rate> Genres
    {
        get => genres;
        set => genres = value;
    }
    public string Title
    {
        get => title;
        set => title = value;
    }
    public string Motto
    {
        get => motto;
        set => motto = value;
    }
    public List<Comment> Comments
    {
        get => comments;
        set => comments = value;
    }
    public List<Actor> Actors
    {
        get => actors;
        set => actors = value;
    }
    public string Devcount
    {
        get => devcount;
        set => devcount = value;
    }

```

```

public string[] Description
{
    get => description;
    set => description = value;
}
public List<Rate> Scores
{
    get => scores;
    set => scores = value;
}
public int Id
{
    get => id;
    set => id = value;
}
public List<Person> Producers
{
    get => producers;
    set => producers = value;
}
public double Fees
{
    get => fees;
    set => fees = value;
}
public List<Person> Directors
{
    get => directors;
    set => directors = value;
}
public double Avscore
{
    get => avscore;
    set => avscore = value;
}
public string Dataprem
{
    get => date;
    set => date = value;
}
public string Agelevel
{
    get => agelevel;

```

```

        set => agelevel = value;
    }
    public double Budget
    {
        get => budget;
        set => budget = value;
    }
    public double Duration
    {
        get => duration;
        set => duration = value;
    }
    public string Fimage
    {
        get => fimage ;
        set => fimage = value;
    }
}
}

```

Б.4 Текст файлу Person.cs

```

using System.IO;

namespace FilmRecomModel
{
    public class Person
    {
        protected string surname;
        protected string name;
        protected int id;

        public Person()
        {
            this.Name = "";
            this.Surname = "";
            this.Id = 0;
        }
        public string Surname
        {
            get => surname;
            set => surname = value;
        }
    }
}

```

```

    }
    public string Name
    {
        get => name;
        set => name = value;
    }
    public int Id
    {
        get => id;
        set => id = value;
    }
}
}

```

Б.5 Текст файла Recommendation.cs

```

using System.IO;

namespace FilmRecomModel
{
    public class Recommendation
    {
        private int id;
        private User ruser;
        private Film rfilm;
        private DateTime dtime;

        public Recommendation ()
        {
            this.Id = 0;
            this.Rfilm = new Film();
            this.Ruser = new User();
            this.Dtime = new DateTime();
        }
        public User Ruser
        {
            get => ruser;
            set => ruser = value;
        }
        public DateTime Dtime
        {
            get => dtime;

```

```

        set => dtme = value;
    }
    public int Id
    {
        get => id;
        set => id = value;
    }
    public Film RFilm
    {
        get => rfilm;
        set => rfilm = value;
    }
}
}

```

Б.6 Текст файла FilmRecomAdminFilm.cs

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using System.IO;
using FilmRecomModel;

namespace FilmRecom
{
    public partial class FilmRecomAdminFilm : Form
    {
        private Database db = new Database();
        private List<Film> AllFilms = new List<Film>();
        private int CurrFilm = 0, CurrActor = 0, CurrProd = 0, CurrDir
= 0;

        private string IMPATH = "img\\";
        public FilmRecomAdminFilm()
        {
            InitializeComponent();
            this.Size = new System.Drawing.Size(1200, 800);
            this.MaximumSize = new System.Drawing.Size(1200, 800);
        }
        private void FrmFRASStart(object sender, EventArgs e)
        {
            GetAllData();
        }
    }
}

```

```

        if (CurrDir >= 0)
        {
            GetFilm();
            CurrDir = AllFilms[CurrFilm].Directors.Count - 1;
            if (CurrDir >= 0) GetDirectors();
            CurrProd = AllFilms[CurrFilm].Producers.Count - 1;
            if (CurrActor >= 0) GetProducers();
            CurrActor = AllFilms[CurrFilm].Actors.Count - 1;
            if (CurrActor >= 0) GetActors();
        }
    }

    public void NewFilm()
    {
        Film film = new Film();
        AllFilms.Add(film);
        CurrFilm = AllFilms.Count - 1;
    }

    void GetFilm()
    {
        tbTitle.Text = AllFilms[CurrFilm].Title;
        dtDataprem.Text = AllFilms[CurrFilm].Date.ToString();
        tbDevcount.Text = AllFilms[CurrFilm].Devcount;
        tbMotto.Text = AllFilms[CurrFilm].Motto;
        tbGenre.Text = "";
        for(int i = 0; i < AllFilms[CurrFilm].Genres.Count; i++)
        {
            tbGenre.Text += AllFilms[CurrFilm].Genres[i];
            tbGenre.Text += ", ";
        }
        tbBudget.Text = AllFilms[CurrFilm].Budget.ToString();
        tbFees.Text = AllFilms[CurrFilm].Fees.ToString();
        cbAgelevel.Text = AllFilms[CurrFilm].Agelevel;
        tbDurt.Text = AllFilms[CurrFilm].Duration.ToString();
        rtbDescription.Lines = AllFilms[CurrFilm].Description;
        tbFimage .Text = AllFilms[CurrFilm].Fimage ;
    }

    public void NewProd()
    {
        Person person = new Person();
        AllFilms[CurrFilm].Producers.Add(person);
        CurrProd = AllFilms[IND].Producers.Count - 1;
    }

```

```

    }

    void GetProducers()
    {
        tPdName.Text =
AllFilms[CurrFilm].Producers[CurrProd].Name;
        tPdSName.Text =
AllFilms[CurrFilm].Producers[CurrProd].Surname;
    }

    public void NewDir()
    {
        Person person = new Person();
        AllFilms[CurrFilm].Directors.Add(person);
        CurrDir = AllFilms[CurrFilm].Directors.Count - 1;
    }

    void GetDirectors()
    {
        tDirName.Text =
AllFilms[CurrFilm].Directors[CurrDir].Name;
        tDSurName.Text =
AllFilms[CurrFilm].Directors[CurrDir].Surname;
    }

    public void NewActor()
    {
        Actor actor= new Actor();
        AllFilms[CurrFilm].Actors.Add(actor);
        CurrActor = AllFilms[CurrFilm].Actors.Count - 1;
    }

    void GetActors()
    {
        tRole.Text = AllFilms[CurrFilm].Actors[CurrActor].Role;
        tActSName.Text =
AllFilms[CurrFilm].Actors[CurrActor].Surname;
        tActImp.Text = AllFilms[CurrFilm].Actors[CurrActor].Photo;
    }

    public void GetAllData()
    {
        foreach (Film f in db.GetAllFilms())

```

```

    {
        AllFilms.Add(f);
        CurrFilm = AllFilms.Count - 1;
    }
}

void FreeFilm()
{
    tbTitle.Clear();
    dtDataprem.Text = "";
    tbDevcount.Clear();
    tbDurt.Clear();
    tbMotto.Clear();
    cbGenre.Text = "";
    tbGenre.Clear();
    tbBudget.Clear();
    tbFees.Clear();
    cbAgelevel.Text = "";
    tbFimage .Clear();
    rtbDescription.Clear();
    tRole.Clear();
    tActSName.Clear();
    tActImp.Clear();
    tPdName.Clear();
    tPdSName.Clear();
    tDirName.Clear();
    tDSurName.Clear();
    pAct.Image = Image.FromFile(IMPATh + "free.png");
    pFimage.Image = Image.FromFile(IMPATh + "free.png");
}

void SaveData()
{
    db.SaveAllFilms(AllFilms);
}

private void rtbDescriptionTChang(object sender, EventArgs e)
{
    try
    {
        AllFilms[CurrFilm].Description = rtbDescription.Lines;
    }
    catch (Exception e6) { }
}

```

```

    }

    private void btnExitClick(object sender, EventArgs e)
    {
        this.Hide();
        Start start = new Start();
        start.Left = this.Left;
        start.Top = this.Top;
        start.Show();
    }

    private void btnNewProdClick(object sender, EventArgs e)
    {
        try
        {
            NewProd();
        }
        catch (Exception e3) { MessageBox.Show(e3.Message); }
        tPdSName.Text = "";
        tPdName.Text = "";
    }

    private void btnSetClick(object sender, EventArgs e)
    {
        bool res = false;
        try
        {
            SaveData();
        }
        catch (Exception e2)
        {
            MessageBox.Show(e2.Message);
            res = true;
        }
        if (!res)
        {
            MessageBox.Show("Дані          фільму          \"\"          +
AllFilms[CurrFilm].Title + "\" успішно внесено в базу даних");
        }
    }

    private void btnProdBeforeClick(object sender, EventArgs e)
    {

```

```

        try
        {
            CurrProd -= 1;
            GetProducers();
        }
        catch(Exception) { CurrProd += 1; }
    }

    private void tRoleTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Actors[CurrActor].Role
tRole.Text;
        }
        catch (Exception e5) { }
    }

    private void btnProdNextClick(object sender, EventArgs e)
    {
        try
        {
            CurrProd += 1;
            GetProducers();
        }
        catch (Exception) { CurrProd -= 1; }
    }

    private void btnBeforeClick(object sender, EventArgs e)
    {
        try
        {
            CurrFilm = CurrFilm - 1;
            GetFilm();
            CurrProd = AllFilms[CurrFilm].Producers.Count - 1;
            GetProducers();
            CurrDir = AllFilms[CurrFilm].Directors.Count - 1;
            GetDirectors();
            CurrActor = AllFilms[CurrFilm].Actors.Count - 1;
            GetActors();
            CurrGenre = AllFilms[CurrFilm].Genres.Count - 1;
            GetGenres();
        }
    }

```

```

        catch (Exception e6)
        {
            CurrFilm = CurrFilm + 1;
        }
    }

    private void btnActBeforeClick(object sender, EventArgs e)
    {
        try
        {
            CurrActor -= 1;
            GetActors();
        }
        catch (Exception) { CurrActor += 1; }
    }

    private void tbBudgetTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Budget
Convert.ToDouble(tbBudget.Text);
        }
        catch (Exception e6) { }
    }

    private void btnActNextClick(object sender, EventArgs e)
    {
        try
        {
            CurrActor += 1;
            GetActors();
        }
        catch (Exception) { CurrActor -= 1; }
    }

    private void tbFeesTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Fees
Convert.ToDouble(tbFees.Text);
        }
    }

```

```

        catch (Exception e6) { }
    }

private void btnNewDirClick(object sender, EventArgs e)
{
    try
    {
        NewDir();
    }
    catch (Exception e3) { MessageBox.Show(e3.Message); }
    tDSurName.Text = "";
    tDirName.Text = "";
}

private void btnDirBeforeClick(object sender, EventArgs e)
{
    try
    {
        CurrDir -= 1;
        GetDirectors();
    }
    catch (Exception) { CurrDir += 1; }
}

private void FRAfrmClose(object sender, FormClosingEventArgs
e)
{
    Application.Exit();
}

private void tbGenreTChang(object sender, EventArgs e)
{
    try
    {
        AllFilms[CurrFilm].Genres = tbGenre.Text;
    }
    catch (Exception e6) { }
}

private void btnNewFilmClick(object sender, EventArgs e)
{
    NewFilm();
    NewProd();
}

```

```

        NewDir();
        NewActor();
        FreeFilm();
    }

private void cbGenreIndChang(object sender, EventArgs e)
{
    if (tbGenre.Text != String.Empty)
        tbGenre.Text += ", ";

    tbGenre.Text += cbGenre.SelectedItem;
}

private void btnDirNextClick(object sender, EventArgs e)
{
    try
    {
        CurrDir = CurrDir + 1;
        GetDirectors();
    }
    catch (Exception) { CurrDir = CurrDir - 1; }
}

private void tPdNameTChang(object sender, EventArgs e)
{
    try
    {
        AllFilms[CurrFilm].Producers[CurrProd].Name
tPdName.Text;
    }
    catch (Exception e5) { }
}

private void btnNewActClick(object sender, EventArgs e)
{
    try
    {
        NewActor();
    }
    catch (Exception e3) {
        MessageBox.Show(e3.Message); }
    tRole.Text = "";
    tActSName.Text = "";
}

```

```

        tActImp.Text = "";
        pAct.Image = Image.FromFile(IMPACTH + "free.png");
    }

    private void tDSurNameTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Directors[CurrDir].Surname =
tDSurName.Text;
        }
        catch (Exception e5) { }
    }

    private void btnNextClick(object sender, EventArgs e)
    {
        try
        {
            CurrFilm = CurrFilm + 1;
            GetFilm();
            CurrProd = AllFilms[CurrFilm].Producers.Count - 1;
            GetProducers();
            CurrDir = AllFilms[CurrFilm].Directors.Count - 1;
            GetDirectors();
            CurrActor = AllFilms[CurrFilm].Actors.Count - 1;
            GetActors();
            CurrGenre = AllFilms[CurrFilm].Genres.Count - 1;
            GetGenres();
        }
        catch (Exception e6) { CurrFilm -= 1; }
    }

    private void tbTitleTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Title = tbTitle.Text;
        }
        catch (Exception e6) { }
    }

    private void tPdSNameTChang(object sender, EventArgs e)
    {

```

```

        try
        {
            AllFilms[CurrFilm].Producers[CurrProd].Surname
tPdSName.Text;
        }
        catch (Exception e5) { }
    }

    private void tbDevcountEd(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Devcount = tbDevcount.Text;
        }
        catch (Exception e6) { }
    }

    private void tDNameTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Directors[CurrDir].Name
tDName.Text;
        }
        catch (Exception e5) { }
    }

    private void tActSNameTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Actors[CurrActor].Surname
tActSName.Text;
        }
        catch (Exception e5) { }
    }

    private void tbDurtTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Duration
Convert.ToInt32(tbDurt.Text);
        }
    }

```

```

        catch (Exception e6) { }
    }

    private void tActNameTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Actors[CurrActor].Name =
tActName.Text;
        }
        catch (Exception e5) { }
    }

    private void tbFimageTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Fimage = tbFimage.Text;
        }
        catch (Exception e6) { }
        try
        {
            pFimage.Image = Image.FromFile(IMPAT
AllFilms[CurrFilm].Fimage );
        }
        catch (Exception e6) { }
    }

    private void tActImpTChang(object sender, EventArgs e)
    {
        try
        {
            AllFilms[CurrFilm].Actors[CurrActor].Photo =
ofdActImp.FileName;
        }
        catch (Exception e5) { }
        try
        {
            pAct.Image = Image.FromFile(IMPAT
AllFilms[CurrFilm].Actors[CurrActor].Photo);
        }
        catch (Exception e5) { }
    }

```

```
private void tbMottoTChang(object sender, EventArgs e)
{
    try
    {
        AllFilms[CurrFilm].Motto = tbMotto.Text;
    }
    catch (Exception e6) { }
}

private void cbAgelevelTChang(object sender, EventArgs e)
{
    try
    {
        AllFilms[CurrFilm].Agelevel = cbAgelevel.Text;
    }
    catch (Exception e6) { }
}

private void dtDatapremTChang(object sender, EventArgs e)
{
    try
    {
        AllFilms[CurrFilm].Date = dtDataprem.Text;
    }
    catch (Exception e6) { }
}

}

}
```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна кваліфікаційна робота магістра

**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
РЕКОМЕНДУВАННЯ ФІЛЬМІВ**

Виконав
студент групи КНТ-129м

П.А. Радєв

Керівник
к.т.н., доцент

В.М. Льовкін

2020

Рисунок В.1 – Слайд 1

Об'єкт дослідження – процес рекомендування фільмів.

Предмет дослідження – методи рекомендування фільмів.

Мета роботи – розроблення програмного забезпечення рекомендування кінофільмів для забезпечення персонального пошуку нових фільмів.

Рисунок В.2 – Слайд 2

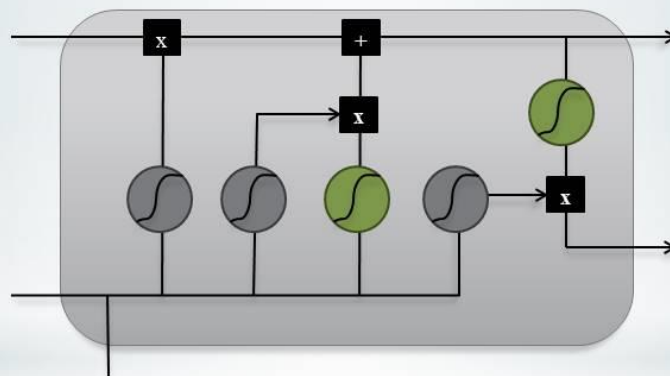
Завдання

- ✓ дослідження методів рекомендування фільмів;
- ✓ розробка технічного завдання;
- ✓ вибір засобів розробки;
- ✓ проектування структури програмного забезпечення;
- ✓ розробка програмного забезпечення рекомендування фільмів;
- ✓ тестування та відлагодження програми;
- ✓ економічне обґрунтування доцільності розробки.

3

Рисунок В.3 – Слайд 3

Нейронна мережа з довгою короткочасною пам'яттю



4

Рисунок В.4 – Слайд 4

Метод рекомендації фільмів

Задача рекомендації фільмів:

$$y = m_{n+1} = f(m_1, m_2, \dots, m_n),$$

де $m_1, m_2, \dots, m_n$ – уподобаний користувачем фільм під номером 1, 2, ..., n відповідно;
 f – функціональна залежність між вхідними ознаками та вихідною ознакою;
 y – вихідна ознака.

На першому етапі має бути реалізовано кодування даних. Для кодування використовується метод word2vec.

На другому етапі має бути виконано підготовку даних. Під час підготовки даних зі всіх даних про уподобання користувачів має бути сформована навчальна вибірка, кожен екземпляр якої представлений наступним чином

$$s_{tr}^o = (x_o, y_o) = ((m_i, m_{i+1}, \dots, m_j), m_{j+1})$$

$$(m_i, m_{i+1}, \dots, m_j) \subseteq S^u,$$

$$S^u = (m_1, m_2, \dots, m_k),$$

де $m_i, m_{i+1}, \dots, m_j, m_{j+1}$ – уподобаний користувачем фільм під номером $i, i+1, \dots, j, j+1$ відповідно;

S^u – множина уподобань фільмів користувача з номером u ;

s_{tr}^o – екземпляр навчальної вибірки з номером o ;

x_o – вхідні ознаки екземпляру навчальної вибірки з номером o ;

y_o – вихідна ознака екземпляру навчальної вибірки з номером o .

На третьому етапі має бути створено та навчено модель на основі ДКП.

На четвертому етапі має реалізовуватися рекомендації фільмів за поточними уподобаннями користувача.

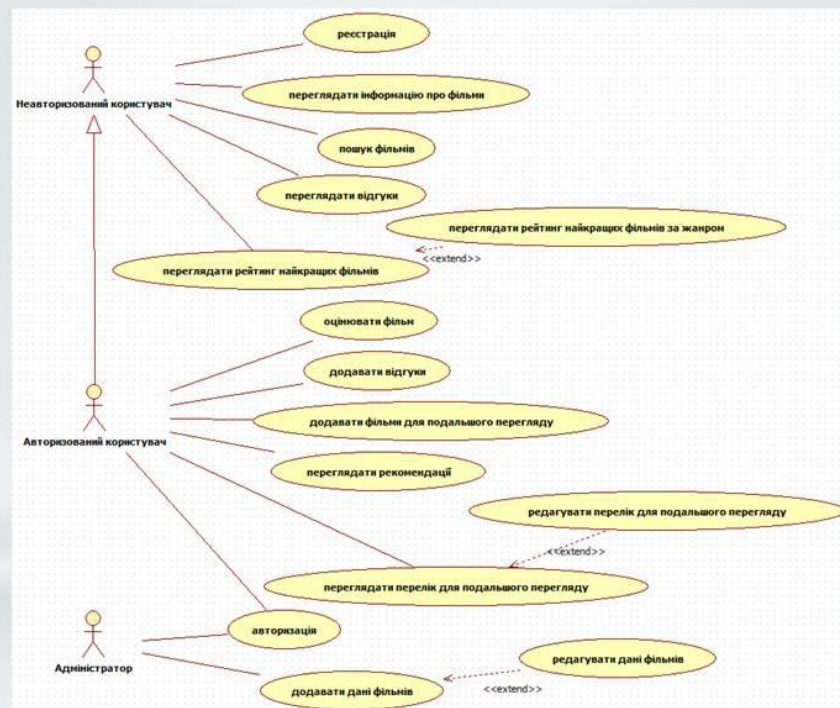
Рисунок В.5 – Слайд 5

Вибір мови програмування

Критерій	C++	C#	Python
Клас мови	компільована, об'єктно-орієнтована (мультипарадигмальна)	компільована, чиста об'єктно-орієнтована (мультипарадигмальна)	інтерпретована, чиста об'єктно-орієнтована (мультипарадигмальна)
Можливості	низькорівневі, високорівневі	високорівневі	високорівневі
Типізація	статична	статична (динамічна)	динамічна
Створення програмних інтерфейсів	абстрактні класи	підтримується напряму мовою	-
Створення Windows-застосунків	+	пряме призначення	можливе
Машинне навчання	можливе	можливе	розширене з широкою підтримкою

Рисунок В.6 – Слайд 6

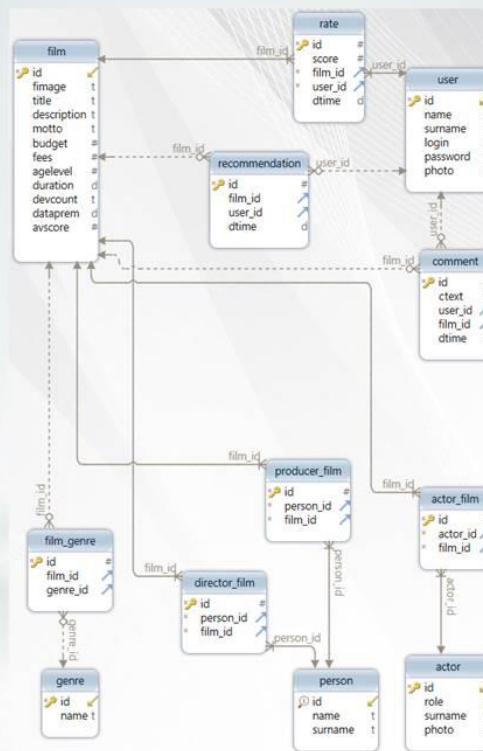
Діаграма прецедентів



7

Рисунок В.7 – Слайд 7

Схема бази даних



8

Рисунок В.8 – Слайд 8

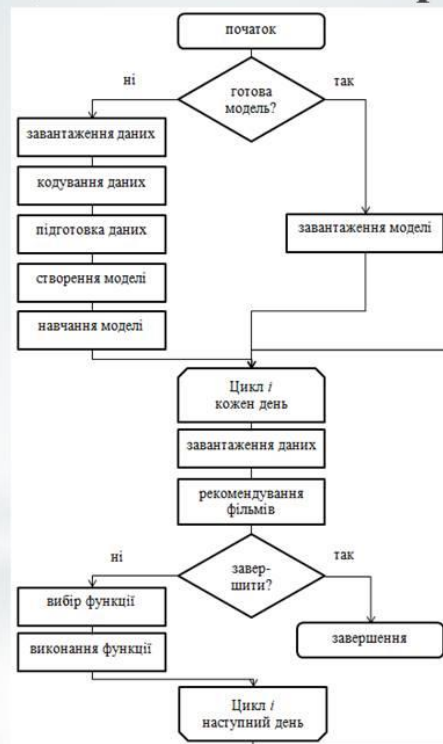
Структурна схема програмної системи



9

Рисунок В.9 – Слайд 9

Функціональна схема програми



10

Рисунок В.10 – Слайд 10

Робота з програмним застосунком

Дослідження та програма реалізації методів рекомендації фільмів

кращі **жанр** всі

рекомендації watchlist **вийти**

знайти

Постер	Назва	Рік	Країна	Жанр	Оцінка	Вік
	Хрещений батько	1972	США	Драма	9,1	R
	Інтерстеллар	2014	США, Велика Британія	Драма, Трилер	8,4	16
	Лабіринт Фавна	2006	Мексика, Іспанія, США	Драма, Фантастика	8,5	16
	Гладіатор	2000	США, Велика Британія	Бойовик, Драма, Пригоди	8,1	R
	Амадей	1984	США	Драма, Історичний	7,1	R
						

11

Рисунок В.11 – Слайд 11

Завантаження даних в програмний застосунок

Дослідження та програма реалізації методів рекомендації фільмів



Назва: **Темні часи**

Слоган: **Ніколи не здавайся!**

Виробництво: **Велика Британія**

Жанр: **Трилер** | Драма, Історичний, Трилер

Дата прем'єри: **26.11.2017**

Тривалість (хвилини): **125**

Вікова категорія: **PG-13**

Бюджет (млн.): **30**

Касові збори (млн.): **150,8**

Актори | Режисери | Продюсери

Ім'я: **Джо**

Прізвище: **Райт**

Постер

Опис

Під час Другої світової війни, коли потужний вермахт Адольфа Гітлера бушує по всій Європі, прем'єр-міністр Великої Британії Невіл Чемберлен змушений подати у відставку, рекомендуючи Вінстона Черчілля як свою заміну. Але навіть у перші дні свого перебування лідером країни, на Черчілля чиниться тиск для того, щоб розпочати мирні переговори з Гітлером або боротися напругу зі, здавалося б, непереможною нацистською владою, якою б не була ціна. Яким би складним і небезпечним не було його рішення, Черчіллю нічого не залишається, як блищати в найтемнішу годину країни. During World War II, as Adolf Hitler's powerful Wehrmacht rampages across Europe, the Prime Minister of the United Kingdom, Neville Chamberlain, is forced to resign, recommending Winston Churchill as his replacement. But even in his early days as the country's leader, Churchill is under pressure to commence peace negotiations with Hitler or to fight head-on the seemingly invincible Nazi regime.

12

Рисунок В.12 – Слайд 12

Рекомендації для перегляду фільмів

Дослідження та програма реалізація методів рекомендації фільмів

— □ ×

Рекомендуємо переглянути наступним



Ірландець (2019)

★★★★★ 7,8

споган	"Його історія змінила історію"
виробництво	США
жанр	Драма
дата прем'єри	27.11.2019
режисер	Мартін Скорсезе
продюсер	Мартін Скорсезе, Роберт Де Ніро, Джейн Розенталь, Гастон Павлович, Емма Коскофф
тривалість	209 хв
віковий рейтинг	R
бюджет	159 000 000 \$
збори	1 000 000 \$

додати у watchlist

відгуки

Френк "Ірландець" Ширен - людина, у якої багато на умі. Колишній вищий чиновник профспілки та вбивця навчився вбивати під час служби в Італії протягом Другої світової війни. Тепер він озирається на своє життя та вбивства, що визначали його мафіозну кар'єру, підтримуючи зв'язок із злочинною родиною Буфаліно. Зокрема, роль, яку він зіграв у зникненні свого друга життя Джиммі Гоффа, колишнього президента Міжнародного братства водіїв вантажівок, який загадково зник наприкінці липня 1975 року у віці 62 років.

Frank "The Irishman" Sheeran is a man with a lot on his mind. The former labor union high official and hitman, learned to kill serving in Italy during the Second World War. He now looks back on his life and

Френк
«Ірландець»



Роберт Де
Ніро Фокс

Джиммі
Хоффа



Аль Пачіно
Ллойд

Рассел
Буфаліно



Джо Пеші
Томпсон

13

Рисунок В.13 – Слайд 13

Висновки

У роботі розглянуто проблему рекомендування фільмів. Визначено структуру програмної системи, що складається з підсистеми для організації моделі предметної області, підсистеми графічного інтерфейса користувача, підсистеми рекомендування фільмів та бази даних. Розроблено програмне забезпечення.

Наукова новизна роботи полягає в запропонованому методі рекомендування фільмів, який характеризується виконанням рекомендування фільмів на основі використання довгої короткочасної пам'яті та використання в якості її вхідних ознак закодованих векторних представлень фільмів за допомогою використання методу word2vec, що дозволяє виконувати рекомендування фільмів з врахуванням послідовності уподобань користувача.

14

Рисунок В.14 – Слайд 14