Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

Бакалаврська програма

Рекомендаційні системи фільмів та їх імплементація в iOS застосунку

Текстова частина до дипломної роботи

за спеціальністю «Інженерія програмного забезпечення» - 121

Керівник курсової роботи

Асистент Бітаєва О.В.

_____

(Підпис)

“___” _____ 2024 року

Виконала студентка БП ІПЗ-4

Авраменко К.В.

“___” _____ 2024 року

Київ 2024

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики технологій факультету інформатики

ЗАТВЕРДЖУЮ
Викладач кафедри інформатики ,
канд. фіз-мат. наук, доц. _____ Гороховський С.С.
(підпис)
„____” _____2024р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на дипломну роботу
студентці Авраменко Катерині Вадимівні
факультету інформатики 4 курсу бакаларської програми
ТЕМА: Рекомендаційні системи фільмів та їх імплементація в iOS застосунку

Зміст ТЧ до курсової роботи:
Індивідуальне завдання
Вступ
Теоретичні основи
Використаний підхід до рішення проблеми
Імплементація обраних технологій та створення застосунку
Висновки
Список покликань

Дата видачі „___” _____ 2024 р. Керівник Асистент
Бітаєва О.В.
(підпис)
Завдання отримала Авраменко К.В.
(підпис)

Тема: **Рекомендаційні системи фільмів та їх імплементація в iOS застосунку**

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	23.10.2023	
2.	Огляд літератури за темою роботи.	23.11.2023	
3.	Проведення досліджень	24.11.2023-24.01.2024	
3.	Опис результатів дослідження	28.02.2024	
4.	Аналіз отриманих результатів з керівником,	10.03.2024	
6.	Корегування роботи	10.04.2024	
7.	Створення презентації та написання доповіді.	10.05.2024	
8.	Остаточне оформлення пояснювальної роботи та слайдів.	10.05.2024	
9.	Захист дипломної роботи	27.05.2024	

Студентка Авраменко К.В.

Керівник Асистент Бітаєва О.В.

10.05.2024

Міністерство освіти і науки України....0

Анотація....*5*

Вступ....*6*

Розділ 1: Теоретичні основи....*8*

1.1 Індустрія фільмів та технології пов'язані з фільмами....8

1.2 Рекомендаційні системи: Основи та технології....9

1.3 Колаборативна фільтрація....11

1.4 Контентна фільтрація....12

1.5 Гібридна фільтрація....14

1.6 Розробка та інтеграція додатків для iOS....15

1.7 Тренування моделей в iOS....17

1.8 Висновки до розділу 1....18

Розділ 2. Використаний підхід до рішення проблеми....*19*

2.1 UX дослідження ринку рекомендаційних додатків на iOS....19

2.2 Рекомендаційні додатки: UI дослідження....22

2.3 SwiftUI як обрана бібліотека для створення зручного UI та взаємодії із системою рекомендацій....24

2.4 Колаборативне фільтрування як обраний метод для тренування моделі..26

2.5 Tensorflow: обрана бібліотека для тренування моделі....28

2.6 Взаємодія з сервером як обраний метод для дотреновування моделі....30

2.7 Висновки до розділу 2....33

Розділ 3. Імплементація обраних технологій та створення застосунку....*34*

3.1 Підготовка набору даних для тренування моделі....34

3.2 Аналіз набору даних....36

3.3 Тренування моделі та її інтеграція в застосунок....39

3.4 Створення дизайну застосунку....41

3.5 Застосунок WhatToWatch як результат дослідження 43

3.6 Тестування застосунку 45

3.7 Стратегія розвитку 46

3.8 Висновки до розділу 3 48

Висновки 49

Список покликань 51

Анотація

У цій роботі досліджуються рекомендаційні системи та їх вплив на індустрію кінематографу. Детально оцінюються можливості надання рекомендацій із кожною рекомендаційною системою. Досліджується впровадження рекомендаційної системи для iOS-додатків, зосереджуючись на кіноіндустрії - секторі, що знаходиться на перетині споживання контенту та технологічних інновацій. Дослідження складається з трьох розділів: теоретичне вивчення систем рекомендацій, включаючи колаборативні, контентні та гібридні методи фільтрації; практичне застосування цих теорій при розробці додатку для iOS з використанням SwiftUI та TensorFlow для створення зручного інтерфейсу та ефективної обробки даних; а також реалізація цих технологій у додатку WhatToWatch. Це дослідження спрямоване на покращення користувацького досвіду шляхом надання персоналізованих рекомендацій щодо контенту, демонструючи потенціал рекомендаційних систем у мобільних додатках. Кінцевий продукт є прикладом того, як передові технології обробки та аналізу даних можуть значно покращити персоналізацію контенту, сприяючи розширенню сфери розваг та інтерактивних медіа.

Вступ

У світі, де культура споживання контенту постійно змінюється, індустрія фільмів перебуває на одній із передових інновацій, що вимагає розвитку нових технологій та методик. Індустрія створення фільмів, їх стримінгу та дистрибютингу є однією із найбільш прибуткових у світі.

Рекомендаційні системи, що активно застосовуються в стримінгових сервісах, соціальних мережах та інших платформах, забезпечують користувачам персоналізований доступ до контенту, значно покращуючи їхній досвід користування. Моя дипломна робота фокусується на розробці та імплементації рекомендаційної системи для iOS-додатків, яка використовує сучасні підходи до обробки та аналізу даних.

В першому розділі дослідження присвячено аналізу теоретичних основ рекомендаційних систем та їхнього застосування у кіноіндустрії. Обговорюються основні технології та методології, такі як collaborative filtering, content filtering та hybrid filtering, які дозволяють ефективно визначати преференції користувачів та пропонувати їм контент, що відповідає їхнім інтересам.

Другий розділ зосереджується на практичному застосуванні отриманих знань при розробці застосунку. Описано дослідження користувацького досвіду (UX) та інтерфейсу (UI), вибір SwiftUI як бібліотеки для створення зручного інтерфейсу, а також Tensorflow як інструменту для тренування рекомендаційної моделі. Значну увагу приділено оптимізації взаємодії з сервером для підтримки актуальності даних і дотренування моделей безпосередньо у мобільному додатку.

Третій розділ присвячено імплементації технологій у практичний застосунок. Розроблено детальний аналіз використаного датасету фільмів,

процес створення рекомендаційної системи та її інтеграція в мобільний застосунок. Також створено естетичний дизайн додатку, який сприяє зручному використанню для користувачів. Заключний продукт дипломної роботи, застосунок "WhatToWatch", є яскравим прикладом ефективної рекомендаційної системи, адаптованої під потреби сучасних користувачів iOS.

Метою цієї дипломної роботи є не лише розробка функціонального та привабливого застосунку, але й глибоке розуміння потенціалу рекомендаційних систем у кіноіндустрії. Через теоретичне вивчення та практичне застосування різних видів фільтрації та аналізу даних, робота прагне дослідити дані технології. Дослідження відображає аналіз потреб сучасних користувачів і тенденцій кіноіндустрії, поєднуючи їх з передовими технологіями і методами розробки програмного забезпечення. Мета роботи — не тільки розробка функціонального додатку, але й внесення вкладу у дослідження рекомендаційних систем та їх інтеграції у мобільні застосунки на базі iOS. Саме вони зможуть вдосконалювати особистісний досвід кожного користувача, що стане в нагоді індустрії розваг та інтерактивних медіа в цілому. Завершальний продукт, застосунок "WhatToWatch", являє собою практичне втілення досліджень і навичок, отриманих під час виконання цієї роботи.

Розділ 1: Теоретичні основи

1.1Індустрія фільмів та технології пов'язані з фільмами

За останні кілька років індустрія кіно і телебачення зазнала значних змін. Глядачі тепер можуть насолоджуватися фільмами, серіалами та телевізійними програмами, що транслюються на їхній долоні, в будь-який час і в будь-якому місці.

На даному новому ринку можна ліцензувати контент, який у попередніх поколіннях вже завершив би свій життєвий цикл. Музику, що датується десятиліттями, можна завантажувати альбомами чи піснями або слухати за підпискою. Контент, який можна переглядати і перепродавати, як, наприклад, телевізійні серіали, що припинили своє існування, тепер доступний кожною мовою і в кожній країні світу.

Потокові технології створюють широкі можливості для творців і дистриб'юторів контенту, але експоненціальне зростання всіх видів відео на замовлення (VOD) і музичних потокових платформ ставить перед ними багато нових викликів.

Як це трапляється з багатьма нішами, для вдалих результатів роботи, сучасні технології стають рішенням та необхідністю для усіх компаній, що зв'язані із даною індустрією. Коли фільми почали з'являтись в доступі онлайн, і глядачам уже не потрібно було іти в кінотеатр та витрачати кошти на квиток, це стало переломним етапом для індустрії дистрибуції фільмів. Саме тоді з метою уникнення піратства та поширення продуктів у хорошій якості, були створені перші стримінгові сервіси.

Стримінг став величезним благом для кіноіндустрії, дозволивши більшій кількості людей, ніж будь-коли раніше, отримати доступ до фільмів і телевізійних шоу. Розміщуючи контент на потокових платформах, режисери та студії можуть охопити ширшу аудиторію, ніж будь-коли

раніше. Це дає їм можливість заробляти більше грошей на своїй роботі, а також зробити так, щоб їхні роботи побачили люди по всьому світу.

Після створення стримінгових сервісів вже не працювали раніше актуальні способи реклами та промоції продуктів кіноіндустрії. Окрім цього, тепер ціллю стало не зацікавити глядача продуктом, а втримати глядача якнайдовше на платформі, навіть якщо його власний вибір фільму виявився хибним і йому не сподобався.

Усі ці зміни, які ми спостерігаємо в тому, як люди споживають розважальний контент, стали можливими завдяки 3 основним факторам.

По-перше, готовність людей обирати, що вони хочуть дивитися або слухати, в будь-який час доби. Як наслідок, фіксований телевізійний розклад більше не працює. По-друге, легший доступ до якісного інтернету з достатньою швидкістю, щоб завантажувати контент в реальному часі. І, нарешті, наявність більш досконалих систем, а також рекомендації для користувачів.

1.2 Рекомендаційні системи: Основи та технології

Саме рекомендаційні системи стали чинником, що повністю змінив кінопромисловість як таку. З точки зору бізнесу, чим більше релевантного контенту або фільмів користувач знаходить на певній платформі, тим вища його зацікавленість і, як наслідок, більший дохід. Різні платформи також виявили, що від 35 до 40% доходу приносять лише рекомендації.

Незважаючи на те, що системи рекомендацій обіцяють користувачеві "кастомізацію", насправді вони є алгоритмами. Вони здатні дізнатися, які теми вам подобаються найбільше, і почати рекомендувати більше фільмів, серіалів і пісень на основі цих уподобань.

Завдяки цьому користувачеві більше не потрібно думати про фільми, які він хоче подивитися, або треки, які він хоче послухати. Зрештою, система рекомендує те, що пов'язане з улюбленими опціями. Як наслідок,

досвід людей на стрімінгових платформах стає скоріше пасивним, ніж активним.

Система рекомендацій фільмів або система кінорекомендацій - це заснований на ML підхід до фільтрації або прогнозування кіноуподобань користувачів на основі їхнього попереднього вибору та поведінки. Це вдосконалений механізм фільтрації, який прогнозує можливий вибір фільмів відповідного користувача та його вподобання щодо конкретного об'єкта, так званого фільму.

Базова концепція системи рекомендацій до фільмів досить проста. Зокрема, у кожній рекомендаційній системі є два основні елементи: користувачі та об'єкти. Система генерує прогнози фільмів для своїх користувачів, тоді як елементи - це самі фільми.

Основна мета систем рекомендацій фільмів - відфільтрувати і передбачити лише ті фільми, які відповідний користувач, найімовірніше, захоче подивитися. Алгоритми ML для цих рекомендаційних систем використовують дані про цього користувача з бази даних системи. Ці дані використовуються для прогнозування майбутньої поведінки відповідного користувача на основі інформації з минулого.

Системи рекомендацій фільмів використовують набір різних стратегій і алгоритмів фільтрації, щоб допомогти користувачам знайти найбільш релевантні фільми. Найпопулярніші категорії алгоритмів машинного навчання, що використовуються для рекомендацій фільмів, включають фільтрацію на основі контенту та системи спільної фільтрації. Рекомендаційна система, що включає в себе алгоритми з даних обидвох систем називається гібридною.

Алгоритми досягають різних рівнів точності і вимагають різного часу для отримання рекомендацій. Для підвищення ефективності рекомендаційних систем були запропоновані та використані різні обчислювальні алгоритми. Однак, кожен алгоритм має свої переваги та

недоліки, що дозволяє використовувати системи для задоволення різних потреб на основі їхніх сильних сторін. Щоб зменшити обмеження кожного з них, алгоритми можуть бути об'єднані таким чином, щоб вони працювали краще при наданні рекомендацій.

1.3 Колаборативна фільтрація

Як випливає з назви, ця стратегія фільтрації ґрунтується на поєднанні поведінки відповідного користувача та інших користувачів. Система порівнює і зіставляє ці поведінки для отримання найоптимальніших результатів. Це колаборація кіноуподобань і поведінки багатьох користувачів.

Основним елементом цієї системи рекомендацій фільмів та алгоритму ML, на якому вона побудована, є історія всіх користувачів у базі даних. По суті, спільна фільтрація ґрунтується на взаємодії всіх користувачів системи з об'єктами (фільмами). Таким чином, кожен користувач впливає на кінцевий результат роботи системи рекомендацій на основі ML, тоді як фільтрація на основі контенту для свого моделювання покладається виключно на дані одного користувача. При цьому розглядаються характеристики користувачів і характеристики об'єктів, які користувачі переглядали або шукали раніше.

Алгоритми спільної фільтрації можна поділити на дві категорії:

Колаборативна фільтрація на основі користувача. Ідея полягає в тому, щоб шукати схожі шаблони у вподобаннях фільмів цільового користувача та інших користувачів у базі даних. Загалом, розглядаються приховані характеристики, отримані з матриць оцінок. У системах рекомендацій фільмів рекомендації надаються на основі інформації про користувача і про те, що дивляться інші люди зі схожою інформацією про користувача. Наприклад, колаборативна фільтрація в системах кінорекомендацій обирає демографічні характеристики користувача, такі як вік, стать та етнічна

приналежність. Завдяки цим функціям створюються рекомендації фільмів, які відповідають іншим людям зі схожими демографічними характеристиками та попередньою пошуковою історією користувача. Колаборативна фільтрація страждає від "холодного старту", якщо користувач не ввів жодної інформації, або інформації занадто мало для точної кластеризації. У цих випадках він не знає, що запропонувати. Точність пропозиції також обмежена, оскільки люди зі схожими демографічними характеристиками можуть не мати схожих уподобань.

Колаборативна фільтрація на основі елементів. Основна концепція полягає в пошуку схожих елементів (фільмів), які цільові користувачі оцінюють або з якими вони взаємодіють.

Сучасний підхід до систем рекомендацій фільмів передбачає поєднання обох стратегій для отримання найбільш поступових і чітких результатів.

1.4 Контентна фільтрація

На відміну від колаборативної фільтрації, методи на основі контенту використовують вектори характеристик користувачів і товарів для надання рекомендацій. Фундаментальна відмінність між цими двома підходами полягає в тому, що системи на основі контенту рекомендують елементи на основі особливостей контенту (не потребують даних про інших користувачів, рекомендують нішеві елементи тощо), тоді як колаборативна фільтрація ґрунтується лише на поведінці користувачів і рекомендує елементи на основі користувачів зі схожими патернами (без знань про предметну область, випадковості тощо). Метод фільтрації на основі контенту працює, пропонуючи користувачеві фільми на основі їхнього змісту. Він визнає, що кластеризація в рекомендаціях спільної фільтрації може не збігатися з уподобаннями користувачів. Смаки та вподобання людей зі схожими демографічними характеристиками дуже відрізняються;

те, що подобається людині X, може не збігатися з тим, що любить дивитися людина Y. Щоб вирішити цю проблему, алгоритми фільтрації на основі контенту надають рекомендації на основі змісту фільмів. У рекомендаціях щодо фільмів деяким змістом є ключові персонажі та жанр фільму.

Також основною відмінністю є те, що система потребує дані лише від одного користувача. Отже в даному випадку проблема 'холодного старту' є не настільки великою. Контентна фільтрація для систем рекомендацій фільмів, використовує лише дані, надані про об'єкти (фільми). Ці дані відіграють вирішальну роль і отримуються лише від одного користувача. Даний алгоритм ML, який використовується для цієї стратегії, рекомендує фільми, схожі на вподобання користувача в минулому. Таким чином, схожість у фільтрації на основі контенту генерується на основі даних про минулий вибір фільмів та вподобання лише одного користувача.

Система рекомендацій аналізує минулі вподобання відповідного користувача, а потім використовує цю інформацію, щоб спробувати знайти схожі фільми. Ця інформація доступна в базі даних (наприклад, головні актори, режисер, жанр тощо). Після цього система надає користувачеві рекомендації щодо фільмів. При цьому основним елементом фільтрації на основі контенту є дані лише одного користувача, які використовуються для прогнозування.

Натомість контекстно-базована система рекомендацій є вдосконаленням методу колаборативної фільтрації. Вона припускає, що якщо особа A і особа Б мають однакову думку щодо питання X, то, найімовірніше, ці ж люди матимуть таку ж думку/вподобання/мислення щодо іншого питання Z. Наприклад, якщо обом людям подобаються різдвяні фільми від Netflix, то, найімовірніше, їм також сподобаються різдвяні фільми від Showmax. Метод фільтрації на основі контексту рекомендує елементи зі схожими функціями або характеристиками, оскільки додатки щойно були розширені до іншого контексту. Він робить ті самі пропозиції,

хоча контексти відрізняються. У більшості випадків веб-браузери імпортують закладки та інші налаштування під час переходу з одного браузера на інший. Це означає зміну контексту, оскільки більшість налаштувань та інших елементів імпортуються в новий контекст, а наявні дані використовуються для надання корисних пропозицій. Аналогічно, системи рекомендацій фільмів можуть надавати подібні рекомендації на основі даних з попереднього контексту. Тут варто згадати про контекстно-орієнтовані рекомендаційні системи (CARS), де поняття контексту чітко визначено. CARS адаптуються до точних умов, в яких буде використовуватися рекомендований елемент. У зв'язку з цим CARS може не рекомендувати користувачеві дуже довгий фільм після напруженого робочого дня або запропонувати романтичний фільм, якщо він/вона перебуває в компанії свого партнера.

1.5 Гібридна фільтрація

Гібридна фільтрація застосовує концепції всіх інших, раніше проаналізованих алгоритмів. Він поєднує в собі спільну фільтрацію, фільтрацію на основі контенту та контекстну фільтрацію для подолання проблем кожного методу. Він є кращим, оскільки досягає вищої продуктивності у формуванні пропозицій, а також швидшого часу обчислень. Наприклад, при спільній фільтрації може не вистачати інформації про доменні залежності, тоді як при фільтрації на основі контенту не вистачає інформації про вподобання людей. Поєднання цих методів дозволяє подолати ці проблеми, оскільки для створення рекомендацій використовуються дані про поведінку користувачів і дані про контент.

1.6 Розробка та інтеграція додатків для iOS

Для демонстрації результатів даного дослідження було обрано імплементувати рекомендаційну систему у iOS застосунок, а також дослідити, яким чином поєднати зручний користувацький інтерфейс із рекомендаційною системою фільмів.

Для розробки iOS застосунку, розробники можуть обрати UIKit або SwiftUI від Apple та створювати візуально привабливі інтерфейси, які відображають рекомендації у зручний для користувача спосіб. В даному випадку було обрано SwiftUI через ряд переваг та зручність у розробці користувацького інтерфейсу.

SwiftUI є сучасним інструментом розробки від Apple для створення інтерфейсів у додатках для iOS, macOS, watchOS та tvOS. Цей фреймворк використовує декларативний підхід до опису інтерфейсів, що сприяє простоті читання та написання коду. Swift UI був обраний як основний інструмент для розробки інтерфейсу iOS-додатку для рекомендації фільмів з наступних причин:

- Його синтаксис є простим та зрозумілим, що сприяє легкості кодування та розуміння коду.
- Фреймворк дозволяє легко візуалізувати макет та поведінку користувацького інтерфейсу завдяки декларативному підходу.
- Swift UI інтегрується з іншими фреймворками Apple, такими як Core Data, полегшуючи розробку стабільних та масштабованих додатків.
- Завдяки функції попереднього перегляду, розробники можуть бачити результати внесених змін у коді в реальному часі, що знижує ризик помилок та прискорює розробку.
- Автоматичне оновлення інтерфейсу дозволяє ефективно управляти станом додатка та забезпечувати користувачам негайний зворотній зв'язок.

Хоча UIKit та Storyboards також використовуються для створення інтерфейсів в iOS, Swift UI пропонує вагомі переваги порівняно з цими альтернативами.

Додаток для iOS повинен мати інтуїтивно зрозумілий та цікавий користувацький інтерфейс. Для того, щоб зробити додаток придатним для використання різними аудиторіями, слід також подбати про доступність та локалізацію. Першою мовою для додатку, було обрано англійську, так як даною мовою володіє найбільша кількість користувачів.

Для локального зберігання даних у додатку було використано CoreData, фреймворк від Apple. Core Data - це потужний фреймворк графів та постійності даних, який надається компанією Apple для додатків iOS та macOS. Він дозволяє розробникам керувати даними та забезпечувати ефективний доступ до них та їх зберігання. Для даного дослідження було вибрано Core Data як основний фреймворк для управління даними у проекті через його переваги над іншими альтернативами.

Переваги Core Data:

- Надає високорівневий рівень абстракції над SQLite, що дозволяє ефективно та оптимізовано здійснювати доступ до даних та їх зберігання.
- Core Data надає багатий набір функцій, включаючи об'єктне моделювання, керування версіями та міграцію схем, які полегшують роботу з даними.
- Він пропонує вбудовану підтримку збоїв та попередньої вибірки, що зменшує використання пам'яті та підвищує продуктивність.
- CoreData підтримує зв'язки між об'єктами, що дозволяє моделювати та підтримувати складні структури даних.

Інші варіанти для управління даними в iOS охоплюють SQLite, Realm та Firebase. Кожен з цих варіантів має свої плюси, але CoreData часто

вибирають через її тісну інтеграцію з екосистемою Apple та широкий спектр функціоналу.

1.7 Тренування моделей в iOS

При створенні системи рекомендацій фільмів для iOS-пристроїв важливо врахувати декілька ключових аспектів:

a. Збір та обробка даних: необхідно зібрати різноманітну інформацію про фільми та оцінки користувачів, після чого дані потрібно очистити та перетворити у формат, придатний для навчання та оцінювання моделей машинного навчання.

b. Вибір і навчання моделі: важливо правильно вибрати модель машинного навчання, що відповідатиме вимогам системи.

c. Точне налаштування та оцінка моделі: необхідно тонко налаштувати параметри моделі, а також оцінити її ефективність, використовуючи метрики якості, такі як точність, повнота, відгук чи F1-оцінка. Цей етап забезпечує, що рекомендації моделі будуть точними та відповідними.

d. Інтерфейс та користувацький досвід: інтерфейс додатку повинен бути інтуїтивно зрозумілим, візуально привабливим і показувати рекомендований одяг зручно та ефективно.

Фреймворк CoreML від Apple полегшує інтеграцію моделей машинного навчання у додатки для iOS, підтримуючи різні типи моделей та забезпечуючи легкий процес їх впровадження. CoreML використовує обчислювальні можливості пристроїв iOS для прогнозування в реальному часі, що надає користувачам швидкий та гнучкий досвід.

Однак у цьому проекті замість CoreML вирішено використати TensorFlow, відкриту бібліотеку машинного навчання, що дозволяє з легкістю реалізувати складні моделі навчання. TensorFlow пропонує гнучкість у розробці і масштабуванні алгоритмів, що робить його ідеальним вибором для експериментів із навчанням моделей. Це забезпечує високу

ефективність обробки даних та можливість виконання моделей на різних платформах, включно з мобільними пристроями.

1.8 Висновки до розділу 1

У першому розділі надано глибокий огляд теоретичних аспектів, що впливають на розробку та інтеграцію рекомендаційних систем в індустрії фільмів за допомогою технологій. Розділ розпочинається з аналізу індустрії фільмів та впливу технологій на спосіб створення, дистрибуції та споживання кіно.

Далі, розділ вдосконалює розуміння основних принципів рекомендаційних систем, обговорюється важливість спільного фільтрування, де рекомендації базуються на поведінці схожих користувачів, а також методи контент-орієнтованого фільтрування, що використовують характеристики самого контенту для створення рекомендацій.

Також згадується гібридний підхід, що комбінує обидва методи для покращення точності та ефективності рекомендацій. На прикладі розробки та інтеграції додатків для iOS розкриваються методики вибору інструментів і фреймворків, які оптимізують використання можливостей пристроїв Apple.

Розділ завершується аналізом можливостей для вдалої розробки застосунку для iOS пристроїв, а також можливостей тренування та оптимізації машинного навчання безпосередньо для iOS. Ці дані закладають теоретичну основу для наступних розділів дослідження, надаючи важливі знання для розуміння ключових технологій та методів, які будуть застосовуватись у практичній реалізації рекомендаційних систем в кіноіндустрії.

Розділ 2. Використаний підхід до рішення проблеми

2.1 UX дослідження ринку рекомендаційних додатків на iOS

У сфері мобільних додатків, що стрімко розвивається, користувацький досвід (UX) відіграє ключову роль у визначенні успіху чи провалу проєкту. Особливо це стосується додатків для рекомендацій фільмів, де здатність надавати персоналізований, релевантний та цікавий контент може суттєво вплинути на задоволеність користувачів та їхнє утримання при користуванні додатком. Цей розділ має на меті дослідити практики UX iOS-додатків для рекомендацій до фільмів, надаючи уявлення про ефективність різних стратегій дизайну та взаємодії.

Щоб провести комплексний аналіз існуючих додатків для рекомендацій фільмів для iOS, було обрано три додатки на основі їхньої популярності та особливостей, було вибрано 1 відомий додаток та два посередньо відомих, для більш ефективного аналізу UX: IMDb, Movie-Recommender та Youcine. Дослідження включало евристичні оцінки, аналіз відгуків користувачів з оглядів в AppStore та власне UX-дослідження на цих платформах.

IMDb пропонує величезну базу даних інформації про фільми та користувацького контенту. Його сильні сторони UX включають високоінформативний інтерфейс і потужний механізм рекомендацій, що працює на основі активності користувачів і методів спільної фільтрації. Однак додаток іноді страждає від інформаційного перевантаження, що може перевантажити нових користувачів.

Інтерфейс додатку Youcine більше нагадує додаток для вибору фільму у кінотеатрі, аніж для рекомендацій, хоч і його основною задачею є порекомендувати користувачу фільм для перегляду. Його унікальністю є те, що користувач може налаштувати багато характеристик для бажаної рекомендації, перш ніж робиться сам запит для неї. Серед цих характеристик можна вказати жанр, бажаний мінімальний рейтинг даного

фільму, а також наскільки давно фільм міг вийти. Серед явних мінусів даного UX є те, що користувачу обов'язково потрібно надавати усі дані, для отримання рекомендацій, тоді як це не завжди те, на що кожен захоче витрачати свій час. Окрім цього, для монетизації додатку додано багато реклами, що перериває позитивний користувацький досвід при користуванні додатком. З чого випливає, що такий спосіб монетизації є не дуже вдалим, і варто було б розглянути таокж інші способи, платну підписку, чи зробити додаток платним на платформі AppStore.

Movie Recommender вирізняється тим, що не потребує введення певних характеристик для отримання рекомендацій. Він також не має проблеми 'холодного старту', так як не вимагає вводити фільми, що були переглянуті користувачем раніше. Натомість користувачу достатньо ввести у текстовому форматі, чи озвучити усно приблизний зміст чи ключові слова для фільму, наприклад 'фільм, у якому присутні детективи'. Особливістю додатку є те, що при наданні дозволу до мікрофону, користувач може диктувати ключові слова чи побажання щодо фільму. Після вводу користувач отримує рекомендацію у вигляді одного фільму, для якого підійшли усі ключові слова, що ввів користувач. Такий функціонал виділяє даний додаток з-поміж інших, щоправда так як це єдиний спосіб отримати рекомендацію, може не завжди підійти для користувачів, що шукають більше, ніж одну рекомендацію чи можуть орієнтуватись не на зміст, а на жанр чи інші характеристики.

Порівняльний аналіз показує, що хоча всі три додатки надають цінні рекомендації, їхні підходи до UX суттєво відрізняються.

IMDb вирізняється своїм інформативним інтерфейсом та потужним механізмом рекомендацій, який використовує спільну фільтрацію та аналіз активності користувачів для забезпечення персоналізованих рекомендацій. Однак, нові користувачі можуть відчути інформаційне перевантаження через обширність доступної інформації, що ускладнює навігацію по

платформі. На відміну від IMDb, Youcine пропонує більш інтуїтивно зрозумілий інтерфейс, з можливістю детально налаштувати параметри пошуку, як жанр чи мінімальний рейтинг. Однак, обов'язковість введення всіх даних може відштовхнути користувачів, які не бажають витрачати час на цей процес, а також переривчастий досвід через численну рекламу може погіршити загальне враження від використання додатку. Movie Recommender, у свою чергу, вирізняється своєю простотою та відсутністю необхідності вводити специфічні характеристики чи переглянуті фільми.

Таким чином, кожен додаток має свої сильні сторони, залежно від потреб користувача: IMDb ідеально підходить для глибокого аналізу фільмів, Youcine пропонує більшу кастомізацію пошуку, а Movie Recommender вирізняється своєю простотою і швидкістю отримання рекомендацій.

Відгуки користувачів з App Store свідчать про різний рівень задоволеності. Найчастіше хвалять вичерпний контент IMDb та функціонал Movie Recommender. Критика часто стосується труднощів навігації, реклами та релевантності запропонованого контенту.

Для покращення користувацького інтерфейсу пропонуються наступні покращення:

IMDb: Спростити користувацький інтерфейс та покращити процес реєстрації, щоб допомогти новим користувачам орієнтуватися у великій кількості контенту. Що також значно спростило би навігацію користувачів – це підказки при першому використанні додатку.

Youcine: Замість обов'язковості введення всіх параметрів для пошуку, можна запропонувати користувачам опцію вибору лише декількох ключових характеристик, залишаючи інші параметри опціональними. Це знизить поріг входження для нових користувачів і зробить досвід користування більш приємним. Щоб зменшити негативний вплив реклами на користувацький досвід, Youcine може ввести преміум підписку, яка

видалить рекламу. Це не тільки покращить загальні враження від користування додатком, але й забезпечить додатковий дохід.

Movie Recommender: хоча додаток і вирізняється своєю здатністю швидко надавати рекомендації, він може виявитися обмеженим для користувачів, що шукають більше варіантів. Розширення можливостей додатку для видачі декількох рекомендацій на один запит може підвищити його привабливість. Окрім цього, розвиток алгоритмів машинного навчання, які зможуть аналізувати не тільки ключові слова, але й контекст запитів, може значно покращити точність рекомендацій.

Цей аналіз підкреслює важливість адаптованих UX-стратегій для успіху додатків з рекомендаціями фільмів. Зосереджуючись на орієнтованому на користувача дизайні та постійно вдосконалюючи стратегії взаємодії, ці додатки можуть значно підвищити рівень задоволеності та залученості користувачів.

2.2 Рекомендаційні додатки: UI дослідження

Для створення естетичного та дружнього користувацького інтерфейсу(UI) було проаналізовано додатки, що вже є на ринку, їх недоліки та переваги. Також важливо прийняти до уваги, які кольори частіше використовуються в даній сфері та який UI додає бажання надалі користуватись саме обраним додатком.

Додаток YouCine використовує темний фон з яскравими контрастними кольорами, що сприяє читабельності та зручності для очей, особливо при використанні в умовах з низьким освітленням. Такий підхід до палітри кольорів є вдалим вибором для додатків, орієнтованих на кінозал та перегляд фільмів увечері. Слайдер вибору рейтингу зірковий та простий у використанні, а великий жовта кнопка підтвердження "Yes, please" для отримання рекомендацій є легко впізнаваною та зручною для натискання.

Вступний екран з малюнком, який представляє користувача у кінотеатрі, створює приємну атмосферу та відображає тему додатку. Проте, присутність рекламного банера може відволікати та погіршувати користувацький досвід, особливо якщо він перекриває елементи інтерфейсу або втручається у процес взаємодії.

Екран зі списком фільмів має чистий та сучасний дизайн, що дозволяє легко переглядати популярні фільми. Однак, кнопка "Surprise Me!" може виглядати як частина навігації, а не як інтерактивний елемент, що може спричинити непорозуміння щодо її функції.

Екран з деталями фільму відображає інформацію з чітко розділеними секціями, що робить контент легко сприйнятним. Кольори та шрифти використані таким чином, що текст виокремлюється на тлі, забезпечуючи хорошу читабельність.

Додаток Movie Recommender також пропонує екран з можливістю пошуку за допомогою текстових запитів, що є інноваційною функцією, яка може виявитися корисною для користувачів, котрі хочуть знайти фільми на основі певної теми або сюжету. Однак, червоний колір у виокремленій області зверху може бути занадто агресивним для очей і може не відповідати загальній темній темі додатку.

Екран із завантаженням зображенням кінозалу додає атмосферу та зв'язок із тематикою кіно. Це може створювати очікування та залученість до вмісту додатку, але важливо, щоб такі екрани не були занадто довгими, щоб уникнути знеохочення користувачів через час очікування.

Обидва додатки використовують чистий дизайн, який не перевантажує користувача зайвою інформацією, що дозволяє легше зосередитися на важливому контенті.

Контраст та кольорова палітра: Темний фон з яскравими елементами візуально привабливий і водночас зручний для користування в різних

умовах освітлення. Важливо підтримувати достатній контраст для забезпечення читабельності.

Інтерактивні елементи: Великі та яскраві кнопки викликають у користувача бажання взаємодіяти з ними, але їх роль і місце повинні бути очевидні, щоб не сплутати користувача.

З урахуванням цих висновків, при розробці нового додатку, особливу увагу слід зосередити на створенні інтуїтивно зрозумілого інтерфейсу, що робить процес пошуку та вибору фільмів простим та приємним. Важливо також забезпечити, щоб усі секції, постери для фільму були контрасними із фоном та чітко розділені завдяки іншим елементам інтерфейсу. Для створення власного дружнього інтерфейсу, рекомендовано взяти наступне: мінімалізм і швидкість від Movie Recommender для простоти використання, інтуїтивність та знайомий дизайн від Youcine для комфортної взаємодії. Це дозволить створити інтерфейс, який не тільки забезпечує глибоку функціональність, але й залишається доступним і приємним у використанні.

2.3 SwiftUI як обрана бібліотека для створення зручного UI та взаємодії із системою рекомендацій

Поява SwiftUI на ринку мобільної розробки не є ізольованою подією. Вже проведено багато досліджень переваг інтерактивного програмування з декларативною кодовою базою для розробки користувацьких інтерфейсів.

Дослідження Королівського Технічного Існиституту (KTH, School of Electrical Engineering and Computer Science), що UIKit має перевагу у продуктивності над SwiftUI, загалом випереджаючи його приблизно на 25%, коли кількість компонентів менша за 32, а також демонструючи стійку тенденцію до меншого споживання пам'яті. Ця різниця у продуктивності може бути важливою у контекстах, пов'язаних з додатками соціальних мереж, де навіть незначні затримки можуть порушити користувацький досвід.

SwiftUI розширює такі переваги, як зменшення довжини коду, пришвидшення часу розробки, попередній перегляд у реальному часі та покращена читабельність коду. Тому рішення про вибір між UIKit та SwiftUI має ґрунтуватися на конкретних вимогах конкретного додатку, а не лише на його продуктивності.

Для даної роботи було вибрано Swift UI як основний фреймворк інтерфейсу для iOS-додатку з рекомендаціями фільмів з наступних причин:

Переваги Swift UI:

- Його синтаксис простий і лаконічний, що полегшує читання і написання коду.
- Він надає декларативний синтаксис для побудови користувацьких інтерфейсів, що дозволяє легко міркувати про макет і поведінку інтерфейсу користувача програми.
- Swift UI призначений для безперебійної роботи з іншими фреймворками Apple, такими як Core Data. Ця інтеграція полегшує створення надійних і масштабованих додатків.
- Він надає функцію попереднього перегляду, яка дозволяла бачити інтерфейс в реальному часі під час редагування коду, що прискорює процес розробки та зменшує ймовірність помилок.
- Swift UI пропонує автоматичне оновлення інтерфейсу, що дозволяє легко керувати станом програми та надавати користувачам зворотній зв'язок у режимі реального часу.

Інші альтернативи для створення інтерфейсу в iOS включають UIKit та Storyboards. Хоча ці альтернативи мають свої переваги, Swift UI пропонує значні переваги над цими варіантами.

2.4 Колаборативне фільтрування як обраний метод для тренування моделі

У сфері рекомендаційних систем було розроблено різні методології для покращення користувацького досвіду шляхом надання персоналізованих пропозицій. Серед них колаборативна фільтрація виділяється як особливо ефективний підхід, особливо для випадків з великою кількістю даних про взаємодію без потреби у великих метаданих про об'єкти або користувачів. У цьому підрозділі досліджується обґрунтування вибору колаборативної фільтрації як методу навчання для нашої системи рекомендацій і обговорюються її переваги над іншими методами.

Спільна фільтрація ґрунтується на припущенні, що користувачі, які погодилися в минулому, погодяться і в майбутньому. Цей метод фокусується насамперед на взаємодії між користувачами та об'єктами, такими як рейтинги або історія переглядів, а не на змісті самих об'єктів. Аналізуючи шаблони взаємодій у великій базі користувачів, система може передбачити ймовірність того, що конкретний користувач віддасть перевагу тому чи іншому об'єкту на основі вподобань схожих користувачів. Цей метод є динамічним за своєю суттю, оскільки він адаптується до нової поведінки користувачів і тенденцій, не вимагаючи явної інформації про товари, що робить його надзвичайно універсальним і масштабованим.

Однією з ключових переваг колаборативної фільтрації є її простота та ефективність у середовищах зі складними та великими обсягами даних про взаємодію користувачів. На відміну від фільтрації на основі вмісту, яка потребує детальної інформації про атрибути елементів, колаборативна фільтрація може ефективно функціонувати навіть за відсутності таких даних. Це робить його особливо цінним у сферах, де метаданих об'єктів мало або їх важко закодувати, наприклад, у фільмах, музиці чи інших видах розваг і медіа.

Крім того, колаборативна фільтрація природно обробляє різноманітні смаки. Оскільки вона вчиться на колективних уподобаннях спільноти користувачів, вона може вловлювати тонкі вподобання та нішеві інтереси, які можуть бути проігноровані більш простими системами, що базуються на правилах. Ця здатність моделювати складні моделі взаємодії дозволяє рекомендаційній системі пропонувати не лише популярні товари, але й менш відомі товари, які тісно пов'язані з унікальними вподобаннями користувача, тим самим підвищуючи задоволеність і залученість користувачів.

Ще однією значною перевагою колаборативної фільтрації є її адаптивність. Коли реєструються нові взаємодії, система може постійно оновлювати свої рекомендації, щоб відображати мінливі тенденції та нову інформацію. Така динамічність допомагає підтримувати актуальність і точність рекомендацій з часом. Крім того, спільна фільтрація може бути реалізована за допомогою різних методів, включаючи матричну факторизацію і глибоке навчання, що дозволяє гнучко обробляти різні масштаби даних і складні моделі взаємодії користувачів.

Для порівняння, інші методи, такі як фільтрація на основі контенту або гібридні підходи, передбачають інтеграцію метаданих про товари або користувачів, що може підвищити точність рекомендацій, але також ускладнює навчання та підтримку моделі. Системи на основі контенту вимагають глибокого розуміння і кодування характеристик об'єктів, що не завжди можливо, особливо у випадках з неструктурованими даними, такими як зображення або відео. Гібридні методи, хоча й потужні, але поєднують складнощі як контент-орієнтованих, так і колаборативних підходів, що вимагає складних механізмів для інтеграції різних джерел даних і типів моделей.

Колаборативна фільтрація була обрана як метод навчання для системи рекомендацій в даній роботі через її надійність у середовищах з великою кількістю даних про взаємодію користувача з об'єктом і здатність надавати персоналізовані рекомендації без потреби в детальних метаданих про об'єкт. Адаптивність до нових даних і здатність враховувати складні користувацькі вподобання роблять його кращим за інші методи, особливо в контекстах, де користувацький досвід має першочергове значення. Використовуючи спільну фільтрацію, система гарантує, що рекомендації залишаються релевантними і глибоко персоналізованими, сприяючи залученню та задоволенню користувачів.

2.5 Tensorflow: обрана бібліотека для тренування моделі

TensorFlow - це фреймворк з відкритим вихідним кодом від Google для наукових та чисельних обчислень з використанням графів потоків даних, які позначають модель виконання TensorFlow. Графи потоків даних, що використовуються в TensorFlow, допомагають експертам з ML виконувати більш просунуте та інтенсивне навчання на своїх даних для розробки DL та прогнозування.

Як випливає з назви, TensorFlow включає операції, що виконуються нейронними мережами над багатовимірними масивами даних, тобто потоками тензорів. Вузли на графі потоку відповідають математичним операціям, тобто додаванню, множенню, факторизації матриць і так далі; тоді як ребра відповідають тензорам, які забезпечують зв'язок між ребрами і вузлами - тобто потік даних і потік управління. Таким чином, TensorFlow надає деякі широко використовувані та надійно реалізовані лінійні моделі та алгоритми DL.

Якщо TensorFlow має доступ до GPU-пристроїв, він автоматично розподіляє обчислення між декількома пристроями за допомогою рудного процесу. Таким чином, для використання ядер центрального процесора не

потрібно спеціальної конфігурації. Тим не менш, TensorFlow також дозволяє програмі визначати, які операції будуть виконуватися на якому пристрої, за допомогою розміщення іменних булочок.

Після оцінки моделі, можна її розгорнути , подаючи їй деякі тестові дані. Вибір TensorFlow для створення рекомендаційної системи має кілька ключових переваг:

- Гнучкість та масштабованість: TensorFlow підтримує широкий спектр алгоритмів і моделей, що робить його придатним як для невеликих додатків, так і для великих, складних систем. Його здатність масштабуватися на різних пристроях - від одного процесора до цілих серверних ферм - робить його придатним для проектів, які можуть зростати з часом.
- Багаті бібліотеки та спільнота: TensorFlow надає великі бібліотеки для побудови та навчання моделей машинного навчання, в тому числі тих, що використовуються в рекомендаційних системах. Завдяки активній спільноті та постійному розвитку розробники мають доступ до найновіших інструментів, а також можуть отримати підтримку та ідеї від інших професіоналів.
- Розширені інструменти для навчання та оцінювання: TensorFlow включає такі інструменти, як TensorBoard для візуалізації мережевої архітектури та продуктивності, що може мати вирішальне значення для розробки ефективних систем рекомендацій. Це допомагає в точному налаштуванні моделі та розумінні складних патернів даних.
- Швидші обчислення: Остання версія TensorFlow неймовірно швидка. Наприклад, модель Inception-v3 працює в 7,3 рази швидше на 8 графічних процесорах, а розподілений Inception-v3 працює в 58 разів швидше на 64 графічних процесорах.

Розробка додатку для iOS, який інтегрує систему рекомендацій на основі TensorFlow, включає в себе кілька кроків, які забезпечують

ефективність, зручність та результативність додатку. Фреймворк Core ML від Apple призначений для ефективної інтеграції моделей машинного навчання в додатки iOS. Core ML підтримує моделі, навчені в TensorFlow, після того, як вони будуть перетворені у формат моделі Core ML. Ця інтеграція дозволяє програмам виконувати завдання машинного навчання локально на пристрої. Не зважаючи на високу ефективність моделей, що розроблені з фреймворком CoreML чи конвертовані в даний формат після тренування, використовуючи TensorFlow, у даній роботі такий підхід не підходив через поставлену задачу дотреновування моделі під час роботи додатку. Це стало ключовим моментом при виборі фреймворку для даної роботи, так як CoreML, фреймворк для ML від Apple не дозволяє дотреновувати модель після її безпосереднього експортування чи зберігання на пристрій.

Використання TensorFlow для розробки системи рекомендацій для iOS-додатків поєднує потужні можливості машинного навчання TensorFlow та складним користувацьким інтерфейсом на базі iOS. Таке поєднання дозволяє створювати персоналізовані та динамічні додатки, які можуть значно підвищити рівень залученості та задоволеності користувачів. Дотримуючись практик розробки моделі TensorFlow та дизайну додатків для iOS, є можливість створювати високоякісні, ефективні додатки, які використовують сильні сторони обох платформ.

2.6 Взаємодія з сервером як обраний метод для дотреновування моделі

Безперешкодна інтеграція моделей у практичні системи, орієнтовані на користувача, має важливе значення. Рішення використовувати Flask для обслуговування системи рекомендацій у поєднанні зі стратегією динамічного перенавчання у відповідь на нові дані користувачів ідеально узгоджується з вимогами адаптивності, реагування в реальному часі та практичного розгортання. У даному підрозділі розглядаються причини, що

стоять за цим методологічним вибором, і підкреслюється, як вони підвищують здатність системи підтримувати ефективні та гнучкі рекомендації.

Рекомендаційна система повинна залишатися актуальною і відповідати потребам користувачів, для того аби система надавала якнайбільш персоналізовані рекомендації. Статичні моделі, які не оновлюються на основі нових даних, швидко втрачають свою ефективність. Динамічне перенавчання моделі щоразу, коли реєструються нові взаємодії, дозволяє системі безперервно розвиватися та адаптуватися. Ця безперервна адаптація має вирішальне значення для підтримки точності та актуальності рекомендацій, які вона надає.

Динамічне перенавчання дозволяє моделі оновлювати свої параметри або перенавчатися з нуля на основі нових даних. Цей процес дозволяє системі враховувати найновіші моделі поведінки користувачів, фіксуючи нові тенденції та зменшуючи вплив застарілих шаблонів. Наприклад, якщо користувач починає цікавитися новим жанром фільмів, динамічно перенавчена модель може швидко підлаштуватися і почати пропонувати фільми з цього жанру, підвищуючи таким чином задоволеність користувача.

Крім того, такий підхід допомагає пом'якшити проблему "холодного старту", характерну для нових користувачів або об'єктів, яким бракує історичних даних про взаємодію. Перенавчання моделі на нових даних гарантує, що система може почати пропонувати персоналізовані рекомендації новим користувачам набагато швидше, ніж якби вона покладалася на менш часті пакетні оновлення.

Для отримання даних з натренованої моделі та взаємодії із нею було обрано метод надсилання та отримання даних через сервер. Flask - це легкий, але потужний веб-фреймворк для Python, відомий своєю простотою та гнучкістю, що робить його зразковим вибором для розгортання моделей машинного навчання. Flask дозволяє розробникам швидко створювати

RESTful API, які мають вирішальне значення для додатків, що потребують взаємодії з внутрішніми сервісами, такими як система рекомендацій.

Вибір Flask є особливо вигідним завдяки кільком його особливостям. Простота використання дозволяє легко інтегруватися з TensorFlow та іншими інструментами на основі Python, що спрощує розробку та підтримку системи. Мінімалістична, але гнучка структура Flask дає розробникам контроль над архітектурою та операціями програми без зайвої функціональності.

Незважаючи на свою простоту, Flask досить надійний для управління значними обсягами трафіку і може бути масштабований за допомогою таких інструментів, як Nginx або Apache, якщо це необхідно. Така масштабованість робить його придатним як для середовища розробки, так і для виробничого середовища. Крім того, широка підтримка спільноти Flask надає безліч плагінів, розширень і ресурсів для усунення несправностей, що сприяє швидкій розробці та вирішенню проблем.

Flask також покращує інтерактивність системи та здатність обробляти дані в режимі реального часу. Обслуговування моделі рекомендацій через API Flask забезпечує миттєву взаємодію. Він може обробляти вхідні дані користувачів, відповідно оновлювати модель і надавати оновлені рекомендації швидко і ефективно, пропонуючи динамічний і цікавий користувацький досвід.

Таким чином, поєднання Flask для розгортання та динамічного перенавчання моделі забезпечує систему рекомендацій необхідними інструментами, щоб залишатися ефективною, релевантною та орієнтованою на користувача. Таке налаштування гарантує, що модель адаптується до постійно мінливих даних, одночасно використовуючи можливості Flask для забезпечення надійної, масштабованої та інтерактивної платформи для залучення користувачів.

2.7 Висновки до розділу 2

У Розділі 2 було досліджено різні аспекти, які сприяють створенню додатку для рекомендацій фільмів, орієнтованого на користувача.

Почато із UX-дослідження, заглибившись у ринкові тенденції та існуючі додатки, щоб зрозуміти потреби та очікування користувачів. Ці знання лягли в основу процесу проектування інтерфейсу, щоб виокремити ключові елементи, які роблять інтерфейс зручним для користувача. Обравши SwiftUI, було використали його потужні можливості для створення інтуїтивно зрозумілого інтерфейсу, який безперешкодно взаємодіятиме із системою рекомендацій. З технічної точки зору, колаборативна фільтрація виявилася оптимальним методом для навчання моделі рекомендацій, в першу чергу через її ефективність у прогнозуванні вподобань користувачів на основі колективних даних. Tensorflow запропонував надійні можливості, необхідні для реалізації та вдосконалення моделі машинного навчання. Нарешті, вибір полегшити навчання моделі через взаємодію з сервером гарантував, що застосунок залишатиметься гнучким та зручним у використанні.

Розділ 3. Імплементація обраних технологій та створення застосунку

3.1 Підготовка набору даних для тренування моделі

Під час підготовки набору даних для навчання системи рекомендацій, в даному випадку, для набору даних MovieLens, початковим кроком є завантаження та попередня обробка даних, щоб забезпечити їхній відповідний формат для їх ефективного опрацювання моделлю. У даній роботі використовується набір даних MovieLens, а саме варіант на 100 тис., який включає як рейтинги, так і фільми, розбиті на навчальні підмножини.

Набір даних про рейтинги містить взаємодію користувачів з фільмами, що дає уявлення про вподобання користувачів. Він включає такі елементи, як ідентифікатори користувачів, назви фільмів та виставлені оцінки. Для нашої моделі рекомендацій попередня обробка набору даних оцінок зосереджується на вилученні лише основної інформації: ідентифікаторів користувачів та назв фільмів. Таке спрощення вилучає інші деталі, такі як точні рейтинги або часові мітки, які не потрібні для початкового навчання нашої моделі рекомендацій.

Аналогічно, набір даних про фільми оптимізовано так, щоб зберегти лише назви фільмів. Це важливо, оскільки назва слугує ключовим ідентифікатором для кожного фільму, і саме її модель використовуватиме для вивчення асоціацій між користувачами та фільмами.

Етап попередньої обробки передбачає застосування перетворення до обох наборів даних. Для рейтингів застосовується функція відображення, яка перетворює кожен запис на словник, що містить лише ідентифікатор користувача та назву фільму. Це перетворення робить набір даних сумісним з операціями TensorFlow, які будуть використані для подальшої обробки та навчання моделі.

Для назв фільмів робиться аналогічне відображення, хоча і простіше, оскільки набір даних вже сфокусований на назвах фільмів, і його потрібно лише перевести у відповідний формат для TensorFlow.

Після того, як набори даних відформатовані належним чином, вони використовуються для адаптації словникових шарів TensorFlow. Ці шари мають вирішальне значення, оскільки вони перетворюють категоріальні дані (ідентифікатори користувачів і назви фільмів) на числові індекси, які може обробляти модель. Шар StringLookup використовується як для ідентифікаторів користувачів, так і для назв фільмів, і, адаптуючи ці шари до відповідних наборів даних, модель вивчає словниковий запас користувачів і фільмів, присутніх у даних. Цей крок гарантує, що кожен користувач і фільм можуть бути унікально ідентифіковані та представлені в моделі.

Адаптовані шари словника потім слугують першим шаром у послідовних моделях як для користувачів, так і для фільмів. Ці моделі включають вбудовування шарів, які слідують за індексами словника, перетворюючи ці індекси в щільні векторні представлення. Таким чином, кожен користувач і фільм представлені 64-вимірним вектором, який фіксує латентні фактори, виведені з даних.

Така підготовка наборів даних є критично важливою, оскільки вона безпосередньо впливає на ефективність системи рекомендацій. Завдяки ретельному відбору та перетворенню даних, забезпеченню впливу на модель лише релевантних ознак та створенню надійного методу обробки категоріальних даних, модель добре підготовлена до вивчення змістовних репрезентацій користувачів та фільмів, що є основою для будь-якої системи рекомендацій.

3.2 Аналіз набору даних

У комплексному аналізі набору даних MovieLens можна заглибитись в різні аспекти даних, щоб виявити закономірності, які можуть допомогти у розробці системи рекомендацій. Аналіз включає в себе завантаження набору даних, вивчення його структури та проведення дослідницького аналізу даних, щоб зрозуміти поведінку користувачів і характеристики фільмів. Для даної роботи було обрано датасет зі ста тисячами рейтингів, поміж інших, де є 25 тисяч та 1 мільйон, так як даний розмір є оптимальним для дослідницької діяльності рекомендаційних систем.

По-перше, набір даних завантажується за допомогою бібліотеки pandas на мові Python, що полегшує маніпулювання та аналіз структурованих даних. Набір даних MovieLens 100k зазвичай складається з декількох файлів, але основними з них є u.data, який містить рейтинги, та u.item, який містить список фільмів.

```
import pandas as pd
ratings = pd.read_csv('u.data', sep='\t', names=['user_id', 'movie_id', 'rating', 'timestamp'])
movies = pd.read_csv('u.item', sep='|', encoding='latin-1', names=['movie_id', 'title', 'release_date', 'video_release_date', 'IMDb_URL', 'unknown', 'Action', 'Adventure', 'Animation', 'Children\'s', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western'])
```

Після завантаження даних виконується попередній аналіз для перевірки типів даних та пошуку відсутніх значень. Цей крок гарантує, що подальший аналіз не буде викривленим або помилковим через проблеми з якістю даних.

```
print(ratings.info())
print(movies.info())
```

З результатів даних методів, можна було побачити, що фрейм даних ratings містить цілочисельні дані та мітки часу, без пропущених значень, що свідчить про достовірність даних про залученість користувачів. Фрейм даних фільмів, окрім назв та ідентифікаторів фільмів, містить жанри, закодовані у вигляді двійкових стовпчиків (1 - наявність жанру, 0 - відсутність), що забезпечує багатовимірний профіль для кожного фільму.

Ключовим аспектом аналізу є вивчення розподілу оцінок і розуміння залучення користувачів до різних фільмів. Це досягається шляхом агрегування даних про рейтинги для обчислення середніх оцінок і підрахунку оцінок для кожного фільму.

movie_ratings = ratings.groupby('movie_id').agg(average_rating=('rating', 'mean'), rating_count=('rating', 'count'))

Об'єднуємо з фреймом даних movies для додавання назв фільмів

movie_ratings = movie_ratings.join(movies.set_index('movie_id')['title'])

print(movie_ratings.sort_values(by='average_rating', ascending=False).head())

Результати цих методів дають уявлення про популярність та якість сприйняття фільмів серед користувачів. Високий середній рейтинг у поєднанні зі значною кількістю оцінок вказує на фільми, які користуються загальним визнанням серед користувачів. І навпаки, фільми з високим рейтингом, але малою кількістю відгуків можуть вказувати на нішеві, але високо оцінені фільми.

Подальший аналіз може включати вивчення поведінки користувачів. Вивчаючи кількість оцінок на одного користувача, ми можемо виявити як випадкових, так і дуже активних користувачів.

user_engagement=ratings.groupby('user_id').size().sort_values(ascending=False)

print(user_engagement.head())

Цей аналіз показує, що залученість користувачів значно варіюється: деякі користувачі оцінюють сотні фільмів, тоді як інші - лише кілька. Така варіативність має вирішальне значення для розуміння динаміки взаємодії користувачів з рекомендаційною системою і може впливати на те, як система налаштовується для задоволення потреб різних типів користувачів.

Крім того, дослідження популярності жанрів допомагає зрозуміти тенденції у вподобаннях користувачів щодо фільмів.

genre_popularity = movies.drop(columns=['movie_id', 'title', 'release_date', 'video_release_date', 'IMDb_URL', 'unknown']).sum().sort_values(ascending=False)

print(genre_popularity)

Аналіз жанрів часто виявляє вподобання щодо певних типів фільмів, таких як драми та комедії, які мають тенденцію бути популярнішими за такі жанри, як документальне кіно або нуар. Розуміння цих тенденцій може допомогти налаштувати систему рекомендацій так, щоб вона акцентувала увагу на більш популярних жанрах, а також виявити можливості в менш насичених нішах.

Таким чином, аналіз набору даних MovieLens 100k пропонує багатогранне бачення кінорейтингів. Вивчаючи рейтинги користувачів, атрибути фільмів та жанрові вподобання, можна отримати детальне розуміння як поведінки користувачів, так і популярності контенту. Загалом, такий аналіз даних є важливим у будь-якому дослідження для повного розуміння набору даних, а бібліотека pandas дає повний функціонал для зручного виконання такого аналізу. Важливо передбачити можливі порожні значення для того, аби не виникало помилок при тренуванні моделі, а також аби це не впливало на саму роботу системи рекомендацій на базі даного набору даних.

3.3 Тренування моделі та її інтеграція в застосунок

При розробці додатку для iOS WhatToWatch, що стане результатом дослідження, для рекомендацій фільмів було впроваджено динамічну систему, яка не лише адаптується до вподобань користувачів, але й забезпечує ефективні та релевантні результати завдяки постійному навчанню та оптимізації. Основна функціональність полягає в системі рекомендацій на основі TensorFlow, розміщеній на сервері, що дозволяє регулярно оновлювати та перенавчати модель на основі взаємодії з користувачем.

При першому використанні додаток пропонує користувачам вибрати десять фільмів, які їм подобаються. Ці початкові дані мають вирішальне значення, оскільки вони встановлюють базову лінію вподобань користувача, дозволяючи рекомендаційній моделі генерувати персоналізований набір кіно-рекомендацій. Вибір початку з десяти фільмів забезпечує баланс між необхідністю введення достатньої кількості даних для отримання змістовних рекомендацій і тим, щоб не перевантажувати користувача.

Оскільки користувачі взаємодіють з додатком, вподобавши додаткові фільми, ці вподобання періодично надсилаються на сервер для уточнення рекомендацій. Зокрема, щоразу, коли користувач вподобав п'ять нових фільмів, робиться запит на сервер, де перетреновується модель. Такий підхід гарантує, що система рекомендацій розвивається разом зі зміною смаків користувача, зберігаючи актуальність і підвищуючи задоволеність користувачів з часом.

Рішення керувати механізмом рекомендацій на сервері, а не вбудовувати його безпосередньо в iOS-додаток за допомогою Core ML, зумовлене кількома факторами:

Складність і розмір: Модель рекомендацій, особливо та, яка потребує регулярного перенавчання на нових даних, є складною і може бути

ресурсномісткою. Розміщення моделі на сервері знімає обчислювальне навантаження з пристрою користувача, забезпечуючи швидшу обробку і менше виснаження батареї та апаратних ресурсів пристрою.

Гнучкість у перенавчанні: Управління моделлю на стороні сервера забезпечує більшу гнучкість у перенавчанні та оновленні моделі без необхідності перерозподіляти додаток. Користувачі отримують вигоду від покращень та оновлень миттєво, без будь-яких дій з їхнього боку.

Для управління даними в додатку для iOS ми використовуємо Core Data, щоб зберігати інформацію про вподобання користувачів та отримані рекомендації. Core Data - це фреймворк від Apple для управління об'єктними графами та постійним сховищем, що робить його ідеальним вибором для нашого додатку з кількох причин:

Інтеграція: Core Data легко інтегрується з iOS, забезпечуючи надійну обробку даних та оптимізацію зберігання з коробки.

Ефективність: Він ефективно керує пам'яттю та графом об'єктів, забезпечуючи безперебійну роботу навіть з великими наборами даних, наприклад, зі зростаючим списком уподобань користувачів та рекомендованих фільмів.

Можливість роботи в автономному режимі: Core Data підтримує зберігання даних локально, що дозволяє додатку ефективно працювати навіть в автономному режимі, кешуючи попередньо завантажені рекомендації та вподобання користувачів.

Ця архітектура, що поєднує потужну модель машинного навчання на стороні сервера з ефективним локальним управлінням даними через Core Data, гарантує, що наш додаток для рекомендацій фільмів є одночасно потужним і зручним для користувача. Користувачі отримують персоналізовані, актуальні рекомендації без недоліків у продуктивності, які зазвичай асоціюються зі складними завданнями машинного навчання на мобільних пристроях. Дизайн системи підкреслює оперативність,

масштабованість і забезпечує безперервне навчання - ключові характеристики для підтримки взаємодії в динамічному середовищі рекомендацій щодо контенту.

Також при першому використанні додатку, для сторінки із загальними рекомендаціями, ми надсилаємо запит, щоб отримати найпопулярніші фільми за запитами користувачів. Цей запит надсилається один раз, і при отриманні найпопулярніших фільмів із серверу, ми зберігаємо дані у CoreData. Надалі, для зображення рекомендацій із фільмами з найбільшим рейтингом по категоріях, ми не співпрацюємо із сервером, а натомість отримуємо дані із локальної бази даних. Це забезпечує швидше отримання даних і відповідно швидшу роботу додатка на даній сторінці.

3.4 Створення дизайну застосунку

Для створення інтуїтивно зрозумілого та привабливого інтерфейсу, було проведено попередній аналіз додатків зі схожою тематикою та досліджено концептуальні та практичні аспекти. Додаток складається з трьох основних вкладок, кожна з яких відповідає окремим потребам користувача: "Персональні рекомендації", "Найкращі фільми за категоріями" та "Фільми, що сподобались".

Вкладка "Персональні рекомендації" - це серце програми. Натхненна популярною механікою свайпів у додатках для знайомств, таких як Tinder, ця функція додає інтерактивний та цікавий елемент до рейтингу фільмів. Користувачі бачать стопку карток фільмів і можуть провести пальцем вправо, якщо фільм їм сподобався або вони його подивилися, і вліво, якщо ні. Така взаємодія не лише робить процес оцінювання цікавим, але й швидким та звичним, використовуючи відомий жест для забезпечення простоти використання.

Щоб покращити користувацький досвід, особливо під час початкового налаштування, було впроваджено індикатор прогресу внизу екрана. Ця

візуальна підказка показує користувачам, скільки фільмів з перших десяти вони вже оцінили. Індикатор прогресу супроводжується заохочувальними фразами, які з'являються з кожним свайпом. Ці повідомлення покликані мотивувати користувачів, створюючи позитивний користувацький досвід, завдяки чому завдання оцінити десять фільмів може більше залучати користувачів та сприяти тому, аби користувач продовжував використання застосунку.

Після того, як користувач оцінив необхідні десять фільмів, додаток використовує ці вподобання для створення персоналізованих рекомендацій щодо фільмів. Ця віддача за початкові зусилля користувача заохочує його до подальшої взаємодії з додатком, що підвищує цінність його вкладу.

Друга вкладка, "Найкращі фільми за категоріями", пропонує просту навігаційну структуру, що дозволяє користувачам швидко знаходити фільми, які їх цікавлять. У кожній категорії представлено список найкращих фільмів, що спрощує процес пошуку. Важливою особливістю цієї вкладки є можливість вподобати фільм подвійним натисканням на його картку - інтуїтивно зрозумілим і швидким жестом. Така взаємодія не лише дозволяє користувачам легко висловлювати свої вподобання, але й покращує збір даних про вподобання користувачів, що, в свою чергу, вдосконалює алгоритм персоналізації.

Нарешті, вкладка "Улюблені фільми" слугує особистим сховищем, де користувачі можуть повернутися до своїх улюблених фільмів. Ця вкладка виконує просту, але важливу функцію: вона дозволяє користувачам легко отримувати доступ до свого списку вподобаних фільмів і керувати ним, гарантуючи, що їхні улюблені фільми завжди будуть на відстані кількох дотиків.

Дизайнерські рішення, прийняті під час розробки цього додатка - від механіки свайпів у вкладці "Особисті рекомендації" до інтуїтивно зрозумілої функції подвійного дотику у вкладці "Найкращі фільми" - були

спрямовані на створення зручного, приємного та ефективного досвіду пошуку фільмів. На основі раніше проведеного аналізу було прийнято рішення обрати мінімалістичний дизайн, без обтяжування жодної зі сторінок зайвими елементами, які б могли візуально перевантажити сторінку чи відволікати увагу користувачів від потрібних елементів. Врахувавши знайомі шаблони взаємодії та включивши елементи гейміфікації, було створено дизайн застосунку, який не лише відповідає практичним потребам, але й забезпечує позитивний користувацький досвід.

3.5 Застосунок WhatToWatch як результат дослідження

Кульмінацією цього дослідження стала розробка застосунку для iOS під назвою WhatToWatch, який використовує модель машинного навчання для надання рекомендацій фільмів. Додаток складається з трьох основних екранів: "Персональні рекомендації", "Найкращі фільми за категоріями" та "Фільми, що сподобались", використовує Core Data від Apple для локального збереження даних та отримує та надсилає дані на сервер для співпраці із моделлю. Створення додатку для рекомендацій фільмів для iOS ґрунтувалося на всебічному дослідженні як поведінки користувачів, так і технологічних рішень. Основна мета полягала в тому, щоб розробити додаток, який не лише рекомендує фільми на основі вподобань користувача, але й робить це за допомогою зручного застосунку.

Спочатку дослідження було зосереджене на розумінні очікувань користувачів від додатку для рекомендацій фільмів. Проаналізувавши дані з існуючих додатків, було визначити ключові функції, які залучають та утримують користувачів. Одним із ключових висновків стало те, що користувачі віддають перевагу персоналізованому та динамічному досвіду, коли рекомендації змінюються відповідно до смаків користувача. Це розуміння стало наріжним каменем дизайну нашого додатку, особливо вплинувши на розробку вкладки "Персональні рекомендації".

З технологічної точки зору, було вирішено впровадити Core Data для локального зберігання користувацьких вподобань, вподобаних фільмів та налаштувань додатку. Цей вибір був мотивований потребою в надійному та швидкому локальному сховищі, яке могло б обробляти складні моделі даних та надавати офлайн-доступ до даних користувачів.

Для надання рекомендацій щодо фільмів було інтегровано модель машинного навчання, розміщену на сервері Flask. Таке налаштування дозволяє нашому додатку взаємодіяти з потужним бекендом, який обробляє дані користувачів, щоб генерувати персоналізовані пропозиції щодо фільмів. Flask був обраний за його простоту та гнучкість, що дозволило нам швидко розробити та розгорнути легкий веб-сервер, який взаємодіє з нашою моделлю машинного навчання. Сама модель навчається на великому наборі даних про рейтинги фільмів та демографічні дані користувачів, щоб з високою точністю передбачати вподобання користувачів.

Зв'язок між додатком WhatToWatch для iOS та сервером Flask здійснюється за допомогою чітко визначених API. Ці API забезпечують безпечну та ефективну передачу даних, гарантуючи захист даних користувачів і дозволяючи оновлювати рекомендації в реальному часі на основі останніх взаємодій користувачів.

Тестування ендпоінтів та взаємодії із сервером відбувалось за допомогою Postman. Ця універсальна платформа дозволила імітувати клієнт-серверну взаємодію, надсилаючи різні HTTP-запити до сервера Flask, гарантуючи, що всі кінцеві точки реагували, як очікувалося. Було ретельно перевірено кожну кінцеву точку на правильність кодів статусу HTTP, час відгуку і точність корисного навантаження. Даний інструмент дає можливості налаштовувати заголовки і змінювати основний вміст, щоб імітувати різні сценарії користувачів. Це дозволило перевірити стійкість і надійність відповідей нашого сервера за різних умов, наприклад, при

обробці некоректного введення даних або неповних параметрів запиту. Здатність Postman зберігати запити для подальшого використання також спрозує процес регресійного тестування, дозволивши швидко проводити повторне тестування після внесення коригувань. Таке ретельне тестування забезпечує стабільність, безпеку та готовність бекенду застосунку до безперебійної взаємодії з користувачами, що значно підвищило загальну якість та продуктивність додатку.

Даний застосунок для iOS є практичним застосуванням проведеного дослідження щодо використання рекомендаційних систем для отримання рекомендацій фільмів. Він надає користувачам інтерактивну, зручну платформу для отримання рекомендацій фільмів, а також демонструє потенціал інтеграції моделей машинного навчання в мобільні додатки.

3.6 Тестування застосунку

Юзабіліті-тестування застосунку WhatToWatch для рекомендацій фільмів для iOS передбачало детальний процес, який включав групу з десяти учасників. Цей етап тестування мав вирішальне значення для того, щоб переконатися, що додаток був доступним та зручним у користуванні.

Учасники були відібрані таким чином, щоб представляти широкий спектр потенційних користувачів, включаючи різний вік та технічні навички. Таке розмаїття допомогло оцінити ефективність додатку в різних сценаріях та вподобаннях користувачів. Кожному учаснику було запропоновано взаємодіяти з усіма функціями додатку, особливо зосередившись на механізмах вподобання фільмів, навігації між вкладками та взаємодії з системою рекомендацій.

Однією з головних цілей юзабіліті-тестування було оцінити, наскільки зручним є додаток у використанні. За учасниками спостерігали, як вони взаємодіяли з додатком, щоб виявити будь-які навігаційні проблеми або плутанину. Механізм свайпів на вкладці "Персональні рекомендації"

отримав особливу похвалу за свою інтуїтивність і простоту використання. Однак деякі учасники зазначили, що деякі елементи інтерфейсу, могли б бути більш помітними. Про механізм свайпів був також доданий негативний відгук щодо нерозуміння того, скільки фільмів потрібно оцінити, перш ніж додаток надасть рекомендації. Цей відгук призвів до коригування дизайну інтерфейсу для покращення візуальної чіткості та взаємодії з користувачем.

Загалом, юзабіліті-тестування надало цінну інформацію про сильні сторони та напрямки вдосконалення нашого додатку для рекомендацій фільмів. Отримані відгуки допомогли вдосконалити дизайн і функціонал додатку, що забезпечило високу зручність використання та доступність продукту.

Тестування доступності застосунку WhatToWatch для рекомендацій щодо фільмів було невід'ємною частиною забезпечення доступності дизайну та функціональності додатку для всіх користувачів, включаючи людей з інвалідністю. Особливу увагу було приділено контрастності кольорів, розміру тексту та доступності інтерактивних елементів, таких як кнопки та повзунки. Проблеми, виявлені під час тестування, такі як читабельність тексту для користувачів з вадами зору та зручність навігації в додатку для тих, хто використовує допоміжні технології, були оперативно вирішені. Це призвело до вдосконалення функцій доступності додатку, були налаштовані опції інтерфейсу, що значно збагатили користувацький досвід для людей з інвалідністю

3.7 Стратегія розвитку

Подальша робота може бути спрямована на подальше вдосконалення алгоритму рекомендацій, збільшення набору даних для більш персоналізованих рекомендацій, покращення користувацького інтерфейсу та створення моделі, що буде навчатись на даних із ще більшою вибіркою

фільмів. У майбутньому, стратегія розвитку додатку WhatToWatch буде зосереджена на вдосконаленні його основних функцій з одночасним впровадженням більш орієнтованих на користувача функцій, які ґрунтуються на реалістичних і досяжних цілях. Ключовим доповненням стане впровадження більш надійної аналітичної системи, яка дозволить глибше зрозуміти поведінку користувачів. Це дасть змогу додатку надавати точніші та індивідуальні рекомендації щодо фільмів завдяки розумінню таких патернів, як звички перегляду та тенденції вподобань.

Ще одним значним оновленням стане розширення можливостей соціальної взаємодії в додатку. Планується впровадження функцій, які дозволять користувачам створювати списки улюблених фільмів, улюблених фільмів в окремих жанрах і ділитися ними, бачити, що дивляться їхні друзі, а також, можливо, інтегрувати функцію, яка дозволить користувачам залишати відгуки або оцінки. Ці соціальні компоненти мають на меті збільшити залученість користувачів, сприяючи створенню атмосфери спільноти навколо відкриття та перегляду фільмів.

Також важливо зосередитись на вдосконаленні користувацького інтерфейсу, щоб забезпечити зручність навігації в додатку, незважаючи на появу нових функцій. Це передбачає оптимізацію продуктивності додатку, щоб він безперешкодно справлявся зі зростаючим навантаженням даних, а також покращення користувацького інтерфейсу, щоб він залишався чистим і доступним, особливо з додаванням нових функцій.

Нарешті, вивчивши можливості інтеграції контенту з інших платформам, буде збагачено наповнення застосунку, доступного користувачам. Це може включати ексклюзивні прев'ю, режисерські нарізки або спеціальні видання фільмів, додаючи унікальний контент, який підвищує ціннісну пропозицію застосунку.

Надаючи пріоритет цим реалістичним та ефективним оновленням, застосунок WhatToWatch прагне зміцнити своє місце як платформи для

рекомендацій фільмів, надаючи користувачам безперешкодний та приємний досвід, який змушує їх повертатися до нього.

3.8 Висновки до розділу 3

У цьому розділі був описаний процес аналізу датасету для його легшого подальшого тренування. Також було описано підготовку датасету до тренування, а також те, як отримана модель надалі була інтегрована в застосунок.

Взаємодія з моделлю через сервер було обраним рішенням для того аби мати можливість дотреновувати модель під час роботи застосунку, та могти покращувати застосунок надалі для більш персоналізованих рекомендацій.

Практичним застосуванням дослідження став застосунок WhatToWatch, що надає користувачам інтерактивну, зручну платформу для отримання рекомендацій щодо фільмів.

Висновки

Дослідження, проведене в цій дипломній роботі, завершується всебічним вивченням і практичним застосуванням рекомендаційних систем, спеціально розроблених для кіноіндустрії. Теоретичний фундамент, закладений у першому розділі, розкриває механіку роботи рекомендаційних систем з акцентом на методах спільної роботи та фільтрації на основі контенту, а також на гібридних підходах. Цей фундамент мав вирішальне значення для розуміння нюансів застосування технологій у контексті рекомендацій до фільмів.

Другий розділ переходить від теорії до практики, детально описуючи процес розробки додатку для рекомендацій фільмів, орієнтованого на користувача. Використання SwiftUI та TensorFlow підкреслило ефективність сучасних інструментів у створенні інтуїтивно зрозумілих інтерфейсів та надійних бекенд-систем. Практичні проблеми впровадження цих технологій були вирішені за допомогою взаємодії на стороні сервера, що дозволило динамічно навчати та оновлювати моделі без шкоди для користувацького досвіду.

Нарешті, застосування цих концепцій було реалізовано в додатку WhatToWatch, що є втіленням досліджень і розробок, детально описаних у цій дипломній роботі. Додаток не лише слугує практичним інструментом для любителів кіно, але й є свідченням потенціалу передових рекомендаційних систем у підвищенні рівня залученості та задоволеності користувачів. Постійно вдосконалюючи модель на основі взаємодії та відгуків користувачів, додаток гарантує, що рекомендації залишатимуться актуальними та персоналізованими.

На закінчення, ця робота не тільки розробила функціональний і естетично привабливий додаток, але й зробила свій внесок у більш широке розуміння того, як системи рекомендацій можуть бути оптимізовані для

мобільних платформ в індустрії фільмів. Незважаючи на проблеми з обробкою даних та системною інтеграцією, подальша робота може розширити цей фундамент, досліджуючи складніші алгоритми та ширші набори даних для подальшого підвищення точності рекомендацій та залучення користувачів.

Список покликань

1. Jayalakshmi, S., Ganesh, N., Čep, R. and Senthil Murugan, J. (2022). Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. *Sensors*, 22(13), p.4904. doi:https://doi.org/10.3390/s22134904.
2. Sunilkumar, Chaurasia. (2020). A review of movie recommendation system: Limitations, Survey and Challenges. ELCVIA Electronic Letters on Computer Vision and Image Analysis. 19. 18. 10.5565/rev/elcvia.1232.
3. Авраменко К. . (2023). Використання машинного навчання для створення мобільної рекомендаційної системи для підбору вбрання
4. Ronneling, B. (2023). *http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1789094*. [online] DIVA. Available at: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1789094&dswid=7728 [Accessed 3 May 2024].
5. Zaccone, G. and Karim, M.R. (2018). *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python, 2nd Edition*. [online] *Google Books*. Packt Publishing Ltd. Available at: https://books.google.com.ua/books?hl=uk&lr=&id=zZlUDwAAQBAJ&oi=fnd&pg=PP1&dq=tensorflow+why+to+choose&ots=rlxMHcdn-X&sig=XloyMqV5a-zSYN3w5mdujmDuFKI&redir_esc=y#v=onepage&q=tensorflow%20why%20to%20choose&f=false [Accessed 3 May 2024].