

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016329162

Дата перевірки:
06.06.2024 20:00:28 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
06.06.2024 20:01:09 EEST

ID користувача:
100012353

Назва документа: 2024_М_ПІ_ІПЗм-22-3_Агатін_Є_Л_скорочений

Кількість сторінок: 46 Кількість слів: 6782 Кількість символів: 52155 Розмір файлу: 1.71 MB ID файлу: 1016128634

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

2.74%

Схожість

Найбільша схожість: 0.99% з джерелом з Бібліотеки (ID файлу: 1011338589)

1.47% Джерела з Інтернету

94

Сторінка 48

1.71% Джерела з Бібліотеки

46

Сторінка 48

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

5

Підозріле форматування

13
сторінок

Рисунок А.1 – Результат перевірки на плагіат (рисунок створено самостійно)

ДОДАТОК Б
Слайди презентації

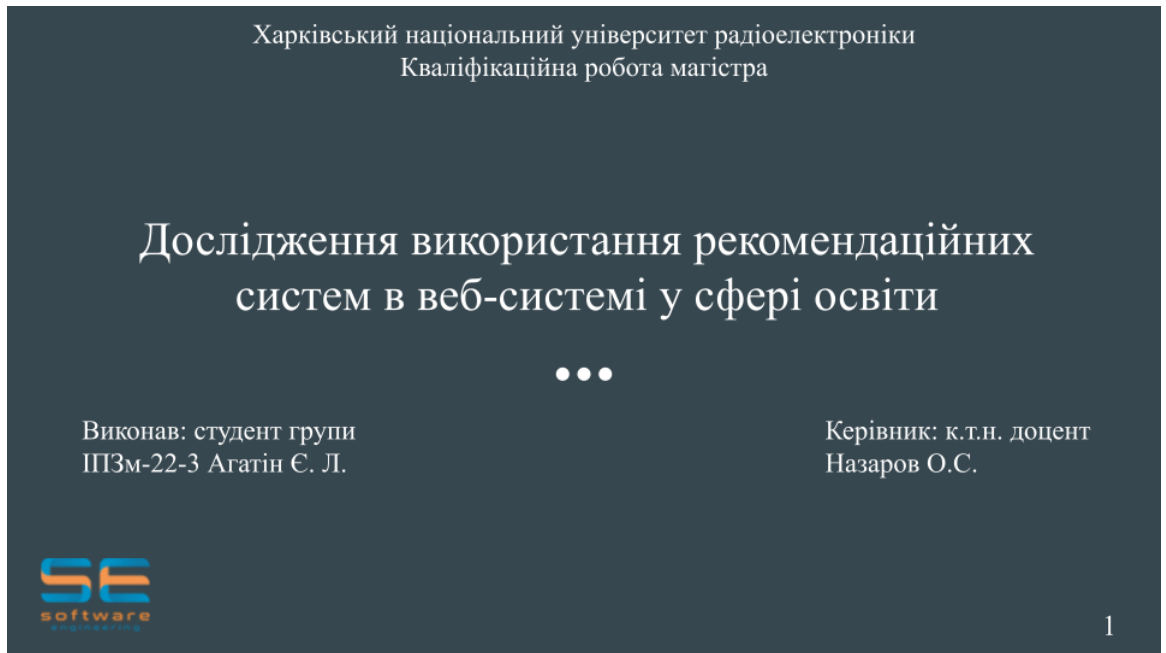


Рисунок Б.1 – Слайд презентації (тема) (рисунок створено самостійно)

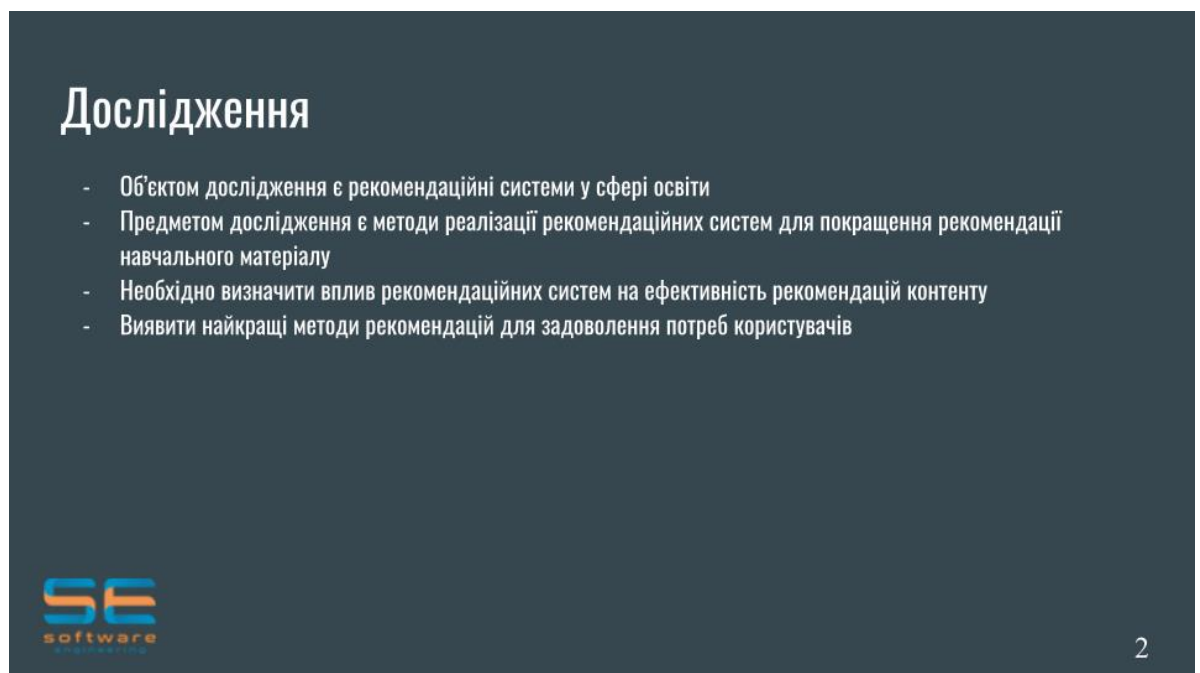



Рисунок Б.2 – Слайд презентації (дослідження) (рисунок створено самостійно)

Постановка задачі

- Проблемою є необхідність забезпечити індивідуалізований підхід та оптимізацію вибору навчальних матеріалів для кожного користувача
- Очікуваним результатом є підвищення якості навчання, персоналізація освітнього процесу, зростання мотивації користувачів та ефективне використання матеріалів




3

Рисунок Б.3 – Слайд презентації (постанова задачі) (рисунок створено самостійно)

Методологія

Використан емпіричний метод дослідження

- Збір даних
- Реалізація алгоритмів
- Експерименти
- Аналіз результатів.



4

Рисунок Б.4 – Слайд презентації (методологія) (рисунок створено самостійно)

Архітектура система для проведення експериментального дослідження

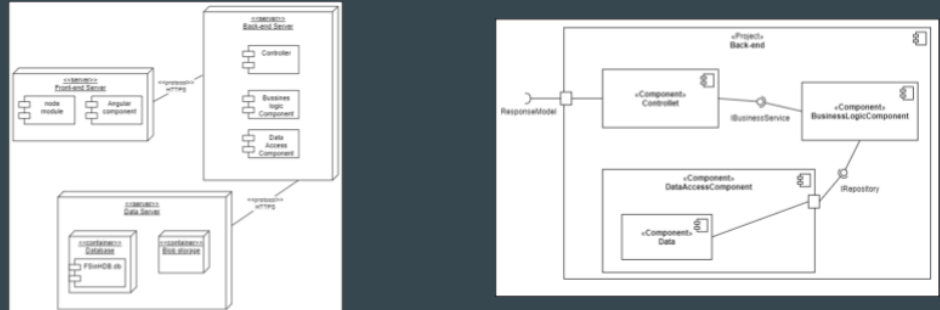


Рисунок Б.5 – Слайд презентації (архітектура) (рисунок створено самостійно)

Схема бази даних

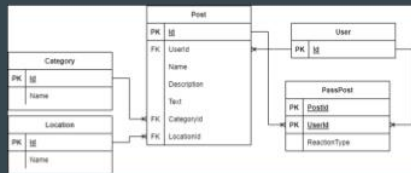


Рисунок Б.6 – Слайд презентації (схема бази даних) (рисунок створено самостійно)

Опис програмного забезпечення, що було використано у дослідженні

- Для роботи з базою даних використовувались: Microsoft SQL Server та середа SQL Server Management Studio Management Studio 19
- Для клієнтської частини: Angular 16, бібліотеки RxJS та бібліотеки стилів Bootstrap, середа Visual Studio Code
- Для серверної частини використовувались: C# 12, бібліотеки .Net 8, також для реалізації алгоритмів використовувались бібліотеки MathNet.Numerics та Microsoft.ML, та середа Visual Studio Code 22



7

Рисунок Б.7 – Слайд презентації (опис інструментів) (рисунок створено самостійно)

Алгоритми

- Колаборативне фільтрування
- Фільтрування на основі вмісту
- Матричне розкладання
- Глибинне навчання
- Обробка природної мови (NLP)



8

Рисунок Б.8 – Слайд презентації (алгоритми) (рисунок створено самостійно)

Вхідні дані

Goodreads Book Graph Datasets

History & Biography (302,935 книжок, 2,066,193 детальних рецензій)

```

{
  "isbn": "1599158683",
  "text_reviews_count": "7",
  "url": "https://www.goodreads.com/book/show/287141.The_Aeneid_for_Boys_and_Girls",
  "authors": [{"author_id": "3841852", "role": ""}],
  "publisher": "Yesterday's Classics",
  "num_pages": "162",
  "publication_day": "13",
  "isbn13": "9781599158683",
  "publication_month": "09",
  "edition_information": "",
  "publication_year": "2000",
  "url": "https://www.goodreads.com/book/show/287141.The_Aeneid_for_Boys_and_Girls",
  "image_url": "https://s.gr-assets.com/assets/hopfoto/book/1111548-bc042a9c31a29c1d080099eff700a03.png",
  "book_id": "287141",
  "ratings_count": "46",
  "work_id": "229219",
  "title": "The Aeneid for Boys and Girls",
  "title_without_series": "The Aeneid for Boys and Girls"
}

```

```

{
  "user_id": "8842281e1d1347389f2ab93d6077344d",
  "book_id": "29893493",
  "review_id": "c23486fb584d6304d1d44c75ce28ea3a",
  "rating": 5,
  "review_text": "I haven't read a non-fiction book this engaging since the story of Nike (or Blue Ribbon Shoes as it was called at the age of 24, to walk into a Japanese shoe manuf start to successful companies (eg Zappos, Amazon, many more to the next, how twice they had to switch banks after being athletes to wear their shoes - often to athletes already we that changed over time. In the story of Prefontaine was a t got a sense of the passion his has for the sport of running frothing, shrieking, the two men entered the final lap. It yard lead, then two, then five. We saw Young grimacing and company." In The book is a fascinating telling of the four more. I could read 4 more volumes of this, as it feels like
  "date_added": "Thu Dec 15 10:51:26 -0800 2016",
  "date_updated": "Sun Mar 12 23:33:51 -0700 2017",
  "read_at": "Thu Mar 09 15:34:06 -0800 2017",
  "started_at": "Tue Feb 28 17:55:35 -0800 2017",
  "n_votes": 29,
  "n_comments": 8
}

```

SE software

9

Рисунок Б.9 – Слайд презентації (вхідні дані) (рисунок створено самостійно)

Які дані розраховуємо

- Точність - співвідношення отриманих правильних результатів до загальної кількості результатів
- Повнота - співвідношення отриманих правильних результатів до загальної кількості очікуваних результатів
- Час - час роботи алгоритму
- Ресурси - необхідний розмір використання пам'яті для роботи алгоритму
- Масштабованість - коефіцієнт зростання часу роботи застосунку від кількості даних

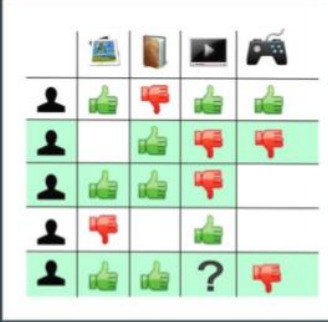
SE software

10

Рисунок Б.10 – Слайд презентації (опис розрахунків) (рисунок створено самостійно)

Колаборативне фільтрування


Схожість з іншими користувачами



```

function getUserSimilarityMatrix(userId) {
  // ...
  for (let i = 0; i < users.length; i++) {
    if (i !== userId) {
      let user = users[i];
      let userItems = user.items;
      let userRatings = user.ratings;
      // ...
    }
  }
  // ...
}

```

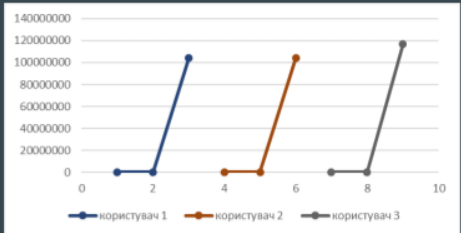



11

Рисунок Б.11 – Слайд презентації (колаборативне фільтрування) (рисунок створено самостійно)

Колаборативне фільтрування результати

користувач	кількість даних	точність	повнота	час	ресурси	масштабованість
1	100	0.223	0.438	14566	415 Mб	206.26
1	500	0.287	0.472	264408	546 Mб	
1	1000	0.248	0.455	104288256	982 Mб	
2	100	0.272	0.483	14242	448 Mб	206.57
2	500	0.215	0.426	263815	504 Mб	
2	1000	0.290	0.490	104054364	906 Mб	
3	100	0.066	0.163	14243	410 Mб	207.63
3	500	0.039	0.142	296239	570 Mб	
3	1000	0.081	0.179	116843094	993 Mб	

12

Рисунок Б.12 – Слайд презентації (колаборативне фільтрування результати) (рисунок створено самостійно)

Матричне розкладання

Лінійне порівняння складових

```
public async Task<IEnumerable<PostItem>> MatrixExpansion(int userId)
{
    var passPosts = await _repository.GetRangeAsync<PassPost>(false, x == true);
    int numUsers = passPosts.Select(x => x.UserId).Distinct().Count();
    int numPosts = passPosts.Select(x => x.PostId).Distinct().Count();

    int i = 0;
    Dictionary<int, int> userMapping = passPosts
        .Select(x => x.UserId).Distinct()
        .Select(x => new { UserId = x, Index = i++ }).ToDictionary(x => x.UserId, x => x.Index);
    i = 0;
    Dictionary<int, int> postMapping = passPosts
        .Select(x => x.PostId).Distinct()
        .Select(x => new { PostId = x, Index = i++ }).ToDictionary(x => x.PostId, x => x.Index);

    double[,] CreateRatingMatrix(IEnumerable<PassPost> passPosts, int numUsers, int numPosts)
    {
        Matrix<double> u, Vector<double> sigma, Matrix<double> vt; PerformSVD(double[,] ratingMatrix);
        double[,] PredictRatings(Matrix<double> u, Vector<double> sigma, Matrix<double> vt);
        double[,] GetRow(double[,] matrix, int rowIndex);
    }

    var ratingMatrix = CreateRatingMatrix(passPosts, numUsers, numPosts);
    var (U, sigma, Vt) = PerformSVD(ratingMatrix);
    var predictedRatings = PredictRatings(U, sigma, Vt);

    var userRatings = GetRow(predictedRatings, userMapping.GetValueOrDefault(userId));
    var recommendations = userRatings
        .Select((rating, index) => new { PostId = index + 1, Rating = rating })
        .OrderByDescending(r => r.Rating)
        .Take(10)
        .Select(r => r.PostId)
        .ToList();

    var allPosts = await _repository.GetRangeAsync<Post>(false, x => recommendations.Contains(x.Id));
    IEnumerable<PostItem> filteredPosts = _mapper.Map<IEnumerable<PostItem>>(allPosts);
    return filteredPosts;
}
```



15

Рисунок Б.15 – Слайд презентації (матричне розкладання) (рисунок створено самостійно)

Матричне розкладання результати

користувач	кількість даних	точність	повнота	час	ресурси	масштабованість
1	100	0.083	0.089	1934	399 Мб	16.04
1	500	0.148	0.119	6542	573 Мб	
1	1000	0.186	0.114	46907	993 Мб	
1	10000	0.132	0.177	1762346	4600 Мб	14.57
2	100	0.044	0.052	3156	412 Мб	
2	500	0.067	0.057	7583	487 Мб	
2	1000	0.101	0.136	28398	962 Мб	15.48
2	10000	0.159	0.198	1066943	4400 Мб	
3	100	0.073	0.097	2220	430 Мб	
3	500	0.134	0.103	7625	575 Мб	15.48
3	1000	0.148	0.179	41597	947 Мб	
3	10000	0.109	0.165	1562844	4300 Мб	



16

Рисунок Б.16 – Слайд презентації (матричне розкладання результати) (рисунок створено самостійно)

Глибинне навчання

На основі аналізу складних взаємодій з публікаціями за допомогою нейронних мереж

```

@Transformer TrainModel(OnContext @Context, @Resource @Resource data)
{
  var dataset = @Context.Data.LoadFromResource(
    data.ResourceId()
    var dataset()
    {
      world = x.world,
      postID = x.postID,
      reaction = int.TryParse(reactionType, out var reactionType)
    }); Take(100);
  var pipeline = @Context.Transform.CsvToJson.Repeat(1000);
  @Resource.Context.Transform.CsvToJson.Repeat(1000);
  @Resource.Context.Transform.CsvToJson.Repeat(1000);
  MatrixColumnColumnNames = "world",
  MatrixColumnColumnNames = "postID",
  LabelColumnNames = "reaction",
  NumHiddenLayers = 10,
  NumHiddenUnits = 100
  });
  var model = @Context.Train(model);
  return model;
}

```

```

public static void Main(string[] args)
{
  @Transformer TrainModel(OnContext @Context, @Resource @Resource data)
  {
    var dataset = @Context.Data.LoadFromResource(
      data.ResourceId()
      var dataset()
      {
        world = x.world,
        postID = x.postID,
        reaction = int.TryParse(reactionType, out var reactionType)
      }); Take(100);
    var pipeline = @Context.Transform.CsvToJson.Repeat(1000);
    @Resource.Context.Transform.CsvToJson.Repeat(1000);
    @Resource.Context.Transform.CsvToJson.Repeat(1000);
    MatrixColumnColumnNames = "world",
    MatrixColumnColumnNames = "postID",
    LabelColumnNames = "reaction",
    NumHiddenLayers = 10,
    NumHiddenUnits = 100
    });
    var model = @Context.Train(model);
    return model;
  }
}

```



Рисунок Б.17 – Слайд презентації (глибинне навчання) (рисунок створено самостійно)

Глибинне навчання результати

користувач	кількість даних	точність	повнота	час	ресурси	масштабованість
1	100	0.238	0.438	3494	429 Mb	0.947
1	500	0.267	0.472	3551	407 Mb	
1	1000	0.285	0.455	2479	397 Mb	
1	10000	0.219	0.483	2794	383 Mb	
2	100	0.252	0.426	2289	444 Mb	1.075
2	500	0.293	0.490	2546	402 Mb	
2	1000	0.276	0.463	2973	361 Mb	
2	10000	0.241	0.442	2812	423 Mb	
3	100	0.262	0.479	2457	442 Mb	1.224
3	500	0.229	0.421	2029	766 Mb	
3	1000	0.278	0.495	3596	419 Mb	
3	10000	0.297	0.458	3857	431 Mb	

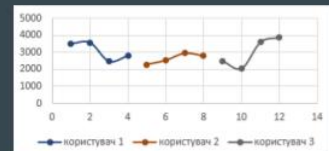


Рисунок Б.18 – Слайд презентації (глибинне навчання результати) (рисунок створено самостійно)

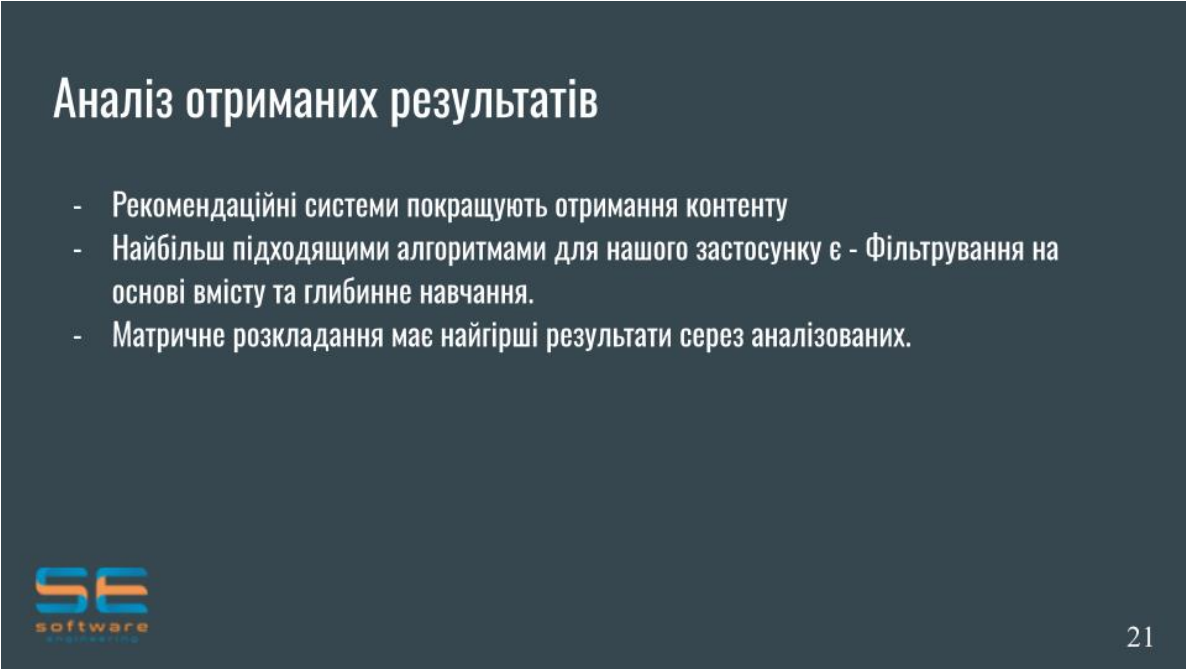


Рисунок Б.21 – Слайд презентації (аналіз отриманих результатів) (рисунок створено самостійно)

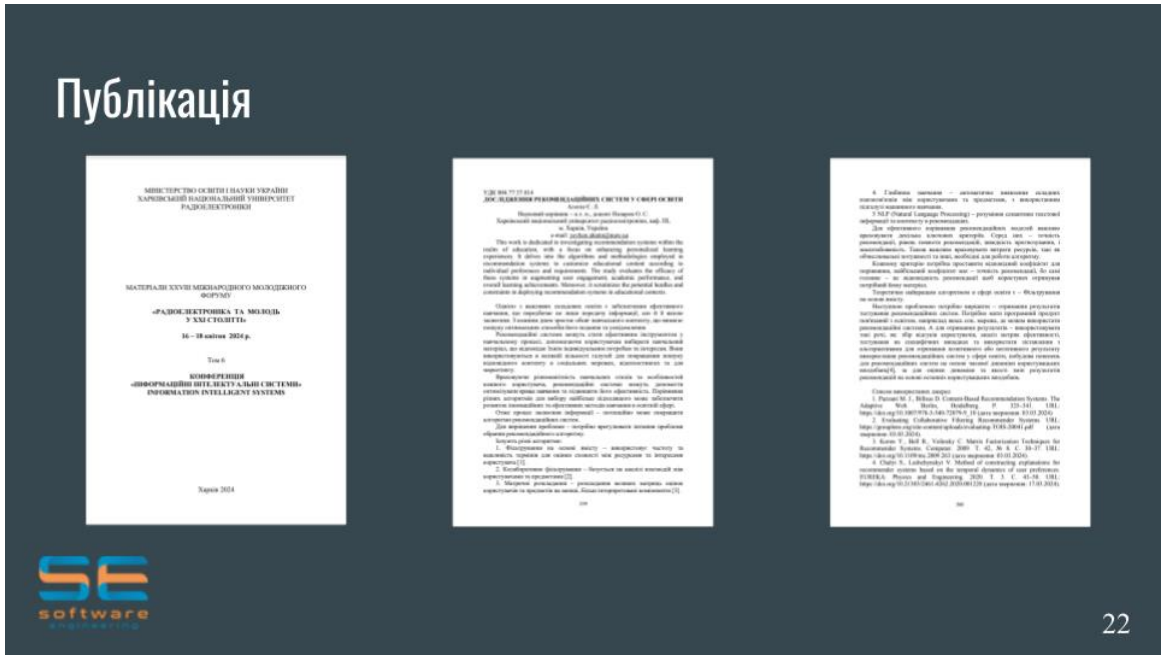



Рисунок Б.22 – Слайд презентації (публікація) (рисунок створено самостійно)

Майбутнє дослідження



- Впровадження застосунку для взяття даних користувачами системи
- Збір зворотних відгуків користувачів для тестування роботи алгоритмів на певних сценаріях
- Аналіз роботи алгоритмів з використанням багатопоточності, кешування, оптимізації специфічних сценаріїв тощо



23

Рисунок Б.23 – Слайд презентації (майбутнє дослідження) (рисунок створено самостійно)

Демо



24

Рисунок Б.24 – Слайд презентації (демонстрація) (рисунок створено самостійно)

Записання



Рисунок Б.25 – Слайд презентації (запитання) (рисунок створено самостійно)

Дякую за увагу

Рисунок Б.26 – Слайд презентації (дякую за увагу) (рисунок створено самостійно)

ДОДАТОК В
Апробація результатів роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ ХХVІІІ МІЖНАРОДНОГО МОЛОДІЖНОГО
ФОРУМУ

**«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ
У ХХІ СТОЛІТТІ»**

16 – 18 квітня 2024 р.

Том 6

**КОНФЕРЕНЦІЯ
«ІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ»
INFORMATION INTELLIGENT SYSTEMS**

Харків 2024

Рисунок В.1 – Перша сторінка збірника (рисунок створено самостійно)

УДК 004.77:37.014

ДОСЛІДЖЕННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У СФЕРІ ОСВІТИ

Агатін Є. Л.

Науковий керівник – к.т. н., доцент Назаров О. С.

Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Україна

e-mail: yevhen.ahatin@nure.ua

This work is dedicated to investigating recommendation systems within the realm of education, with a focus on enhancing personalized learning experiences. It delves into the algorithms and methodologies employed in recommendation systems to customize educational content according to individual preferences and requirements. The study evaluates the efficacy of these systems in augmenting user engagement, academic performance, and overall learning achievements. Moreover, it scrutinizes the potential hurdles and constraints in deploying recommendation systems in educational contexts.

Однією з важливих складових освіти є забезпечення ефективного навчання, що передбачає не лише передачу інформації, але й її якісне засвоєння. З кожним днем зростає обсяг навчального контенту, що вимагає пошуку оптимальних способів його подання та усвідомлення.

Рекомендаційні системи можуть стати ефективним інструментом у навчальному процесі, допомагаючи користувачам вибирати навчальний матеріал, що відповідає їхнім індивідуальним потребам та інтересам. Вони використовуються в великій кількості галузей для покращення пошуку відповідного контенту в соціальних мережах, відеохостингах та для маркетингу.

Враховуючи різноманітність навчальних стилів та особливостей кожного користувача, рекомендаційні системи можуть допомогти оптимізувати процес навчання та підвищити його ефективність. Порівняння різних алгоритмів для вибору найбільш підходящого може забезпечити розвиток інноваційних та ефективних методів навчання в освітній сфері.

Отже процес засвоєння інформації – потенційно може покращити алгоритми рекомендаційних систем.

Для вирішення проблеми – потрібно врегулювати питання проблеми обрання рекомендаційного алгоритму.

Існують різні алгоритми:

1. Фільтрування на основі вмісту – використовує частоту та важливість термінів для оцінки схожості між ресурсами та інтересами користувача [1].
2. Колаборативне фільтрування – базується на аналізі взаємодій між користувачами та предметами [2].
3. Матричні розкладання – розкладання великих матриць оцінок користувачів та предметів на менші, більш інтерпретовані компоненти [3].

Рисунок В.2 – Перша сторінка тез (рисунок створено самостійно)

4. Глибинне навчання – автоматичне виявлення складних взаємозв'язків між користувачами та предметами, з використанням підгалузі машинного навчання.

5 NLP (Natural Language Processing) – розуміння семантики текстової інформації та контексту в рекомендаціях.

Для ефективного порівняння рекомендаційних моделей важливо враховувати декілька ключових критеріїв. Серед них – точність рекомендації, рівень повноти рекомендацій, швидкість прогнозування, і масштабованість. Також важливо враховувати витрати ресурсів, такі як обчислювальні потужності та інші, необхідні для роботи алгоритму.

Кожному критерію потрібна проставити відповідний коефіцієнт для порівняння, найбільший коефіцієнт має – точність рекомендації, бо самі головне – це відповідність рекомендації щоб користувач отримував потрібний йому матеріал.

Теоретично найкращим алгоритмом в сфері освіти є – Фільтрування на основі вмісту.

Наступною проблемою потрібно вирішити – отримання результатів тестування рекомендаційних систем. Потрібно мати програмний продукт пов'язаний з освітою, наприклад якась соц. мережа, де можна використати рекомендаційні системи, А для отримання результатів – використовувати такі речі, як: збір відгуків користувачів, аналіз метрик ефективності, тестування на специфічних випадках та використати зіставлення з альтернативами для отримання позитивного або негативного результату використання рекомендаційних систем у сфері освіти, побудова пояснень для рекомендаційних систем на основі часової динаміки користувацьких вподобань[4], за для оцінки динаміки та якості змін результатів рекомендацій на основі останніх користувацьких вподобань.

Список використаних джерел

1. Pazzani M. J., Billsus D. Content-Based Recommendation Systems. The Adaptive Web. Berlin, Heidelberg. P. 325–341. URL: https://doi.org/10.1007/978-3-540-72079-9_10 (дата звернення: 03.03.2024)
2. Evaluating Collaborative Filtering Recommender Systems. URL: <https://grouplens.org/site-content/uploads/evaluating-TOIS-20041.pdf> (дата звернення: 03.03.2024).
3. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. Computer. 2009. Т. 42, № 8. С. 30–37. URL: <https://doi.org/10.1109/mc.2009.263> (дата звернення: 03.03.2024).
4. Chalyi S., Leshchynskiy V. Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences. EUREKA: Physics and Engineering. 2020. Т. 3. С. 43–50. URL: <https://doi.org/10.21303/2461-4262.2020.001228> (дата звернення: 17.03.2024).

ДОДАТОК Г

Експертний висновок результатів перевірки кваліфікаційної роботи на
відповідність оформлення вимогам ДСТУ 3008: 2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент
(посада)

програмної інженерії
(кафедра)

ІІЗМ-22-3
(група)

Агатін Євген Леонідович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні позначки	
	7.15 Додатки	

зауважень немає

Експерт

(підпис)

Олена ОЛІЙНИК

(прізвище, ініціали)

08.06.2024

Рисунок Г.1 – Результат перевірки на нормоконтроль (рисунок створено
самостійно)

ДОДАТОК Д

Опис алгоритмів за характеристиками

Таблиця Д.1 – Характеристики алгоритмів. (таблицю створено самостійно)

	Точність рекомендацій	Повнота рекомендацій	Швидкість прогнозування
Фільтр заснований на контенті	Висока точність можлива, якщо характеристики ресурсів та інтереси користувачів добре відображені. Однак точність може знижуватися, якщо недостатньо інформації про характеристики або якщо смаки користувачів змінюються.	Може бути обмежена, оскільки рекомендації ґрунтуються на інформації про контент ресурсів. Якщо ресурси недостатньо описані або є нові, фільтр може бути менш ефективний.	Зазвичай висока, оскільки обчислення схожості може виконуватися швидко на основі характеристик ресурсів та профілю користувача.
Фільтр заснований на спільному використанні	Точність може бути високою, особливо якщо є достатньо інформації про взаємодії користувачів. Але може бути менш точним для нових користувачів або ресурсів.	Може бути добре високою, оскільки рекомендації базуються на історії взаємодії користувачів. Проте для нових користувачів або ресурсів може бути менш ефективним.	Зазвичай висока, оскільки обчислення схожості між користувачами може бути виконано швидко.
Матричні розкладення	Матричні розкладення можуть досягати високої точності, оскільки вони можуть виявити складні взаємодії та патерни в даних.	Залежить від того, наскільки добре матричні розкладення можуть враховувати всі можливі взаємодії.	Швидкість залежить від методу факторизації та оптимізаційних методів..
Глибинне навчання	Глибинне навчання може досягати високої точності, особливо коли використовуються складні архітектури мереж та обширні набори даних.	Може бути ефективним у виявленні складних зв'язків та рекомендацій для різноманітних користувачів та ресурсів.	Швидкість може бути питанням через велику кількість параметрів та глибоку структуру мереж.
Застосування NLP	Застосування NLP може призвести до високої точності, оскільки воно дозволяє враховувати контекст та семантичні зв'язки в текстовій інформації.	Залежить від якості та обсягу текстових даних, які можна використовувати для аналізу.	Великі та складні моделі, такі як глибокі нейронні мережі, можуть вимагати значних обчислювальних ресурсів та часу для прогнозування.

Таблиця Д.2 – Характеристики алгоритмів. (таблицю створено самостійно)

	Масштабованість	Споживач ресурсів
Фільтр заснований на контенті	Залежить від обсягу контенту та кількості користувачів. Масштабований для невеликих до середніх обсягів даних, але може стикатися з обмеженнями для великих обсягів.	Зазвичай має низький споживання ресурсів, оскільки обчислення схожості базуються на аналізі характеристик ресурсів та профілю користувача. Не вимагає значних обчислювальних потужностей чи пам'яті.
Фільтр заснований на спільному використанні	Масштабований для середніх обсягів даних, але може зазнавати обмежень для великих обсягів або при збільшенні кількості користувачів та ресурсів.	Зазвичай має помірне споживання ресурсів. Індексація та обчислення схожості може вимагати обчислювальних ресурсів, особливо для великих наборів даних.
Матричні розкладення	Масштабованість може бути проблемою для великих матриць, оскільки обчислення факторизації може вимагати значних ресурсів.	Матричні розкладення можуть бути витратними з точки зору обчислювальних та пам'ятевих ресурсів, особливо для великих даних. Оптимізація алгоритмів і використання розподілених обчислень можуть полегшити це.
Глибинне навчання	Глибинне навчання може бути обмеженим у випадках великих обсягів даних, оскільки навчання глибоких моделей може вимагати значних ресурсів та часу.	Споживання ресурсів може бути великим, особливо для глибоких моделей та великих даних. Потрібна потужна обчислювальна архітектура та можливо навчальні витрати.
Застосування NLP	Масштабованість може бути викликана обробкою великої кількості текстових даних. Використання розподілених систем обробки даних та оптимізація алгоритмів може допомогти вирішити ці проблеми.	Споживання ресурсів може бути значним, особливо для великих моделей, які використовують глибокі нейронні мережі. Використання високопродуктивних архітектур і оптимізація можуть допомогти зменшити ці витрати.