Інформаційні технології

УДК 004

Єрошенко Олександр Сергійович
студент
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
Ерошенко Александр Сергеевич
студент
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»
Yeroshenko Oleksandr
Student of the
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

РЕКОМЕНДАЦІЙНІ СИСТЕМИ В ДИСТАНЦІЙНОМУ НАВЧАННІ
РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ В ДИСТАНЦИОННОМ ОБУЧЕНИИ
RECOMMENDER SYSTEMS FOR USE IN DISTANCE LEARNING

Анотація. *У дослідженні проаналізовано методи створення та реалізації рекомендаційних систем. Наведено порівняння результатів роботи різних методів. Запропоновано способи впровадження рекомендаційних алгоритмів у систему дистанційного навчання.*

Ключові слова: *система дистанційного навчання, рекомендаційна система, колаборативна фільтрація, фільтрація заснована на вмісті.*

Аннотация. *В исследовании проанализированы методы создания и реализации рекомендательных систем. Приведено сравнение результатов*

работы различных методов. Предложены способы внедрения рекомендательных алгоритмов в систему дистанционного обучения.

Ключевые слова: *система дистанционного обучения, рекомендательная система, коллаборативная фильтрация, фильтрация основанная на содержании.*

Summary. *The research carried out an analysis of the methods of creating and implementing recommender systems. The comparison of the results of various methods is given. The methods of implementation of advisory algorithms in the system of distance learning are offered.*

Key words: *distance learning, recommender system, collaborative filtering algorithm, content-based filtering algorithm.*

Постановка проблеми. Сучасне життя важко уявити без технологій та пристроїв на кшталт комп'ютера та смартфона.

Окрім звичного використання, технології можуть надати нам доступ до різноманітних освітніх матеріалів. Одним із способів взаємодії з цими матеріалами є системи дистанційного навчання.

Системи дистанційного навчання не відрізняються від додатків, де представлені книги, музичні твори, кінофільми, відеоігри тощо. Функція, без якої неможливо уявити інтерфейси таких систем – блок рекомендацій та пропозицій. У цих елементах користувач може бачити продукти, які могли б йому сподобатись. Це можливо завдяки рекомендаційним системам.

Виклад основного матеріалу. Є кілька підходів для створення рекомендаційних систем: використання колаборативної фільтрації, фільтрації заснованої на вмісті та гібридної фільтрації, яка поєднує в собі два інших для приховання їхніх недоліків та наголошенні на перевагах [1].

В усіх цих методах необхідно знаходити подібності елементів. У алгоритмі колаборативної фільтрації знаходяться схожі користувачі, а у алгоритмі фільтрації заснованої на вмісті – схожі елементи.

Розглянемо методи обчислення подібностей докладніше. Для демонстрації будемо знаходити схожих користувачів.

Для початку поглянемо на набір тестових даних (таблиця 1).

Таблиця 1

Тестові дані для надання рекомендацій.

Курси	Олександр	Валерія	Олена	Павло
Бази даних	13	3	11	-
Комп'ютерні мережі	10	-	-	3
Xamarin	6	1	9	-
Фронт-енд	-	6	-	9
UI/UX	-	7	1	8

Для знаходження схожих курсів використовуються схожі дані (таблиця 2).

Таблиця 2

Вхідні дані для знаходження схожих курсів

Користувачі	Бази даних	Комп'ютерні мережі	Xamarin	Фронт-енд	UI/UX
Олександр	13	10	6	-	-
Валерія	3	-	1	6	7
Олена	11	-	9	-	1
Павло	-	3	-	9	8

Надалі будемо працювати з таблицею 1 та надаватимемо рекомендації користувачу Олександр. Також можна зауважити, що користувачі Олександр та Олена дуже схожі. Це нам знадобиться під час аналізу результатів.

Є кілька способів обрахувати схожість користувачів: Евклідова відстань, кореляція Пірсона, манхеттенська метрика та косинусна відстань. Для того, щоб розуміти, як проводяться обрахунки, зазначимо, що користувачі уявно розташовуються на графіку відносно їхніх оцінок. На рисунку 1 показано, як можна схематично представити користувачів в залежності від їхніх оцінок курсам «Бази даних» та «Xamarin».

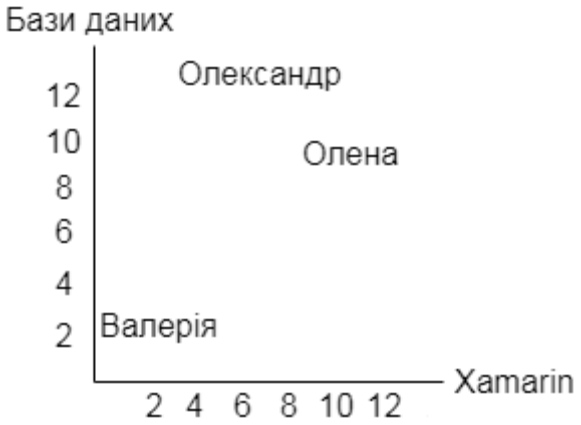

Рис. 1. Графік вподобань користувачів

Евклідова відстань – це дуже простий спосіб розрахунку оцінки подібності. Результат обчислень подібностей за допомогою власної реалізації та функції euclidean_distances з бібліотеки sklearn.metrics.pairwise мови Python наведений у таблиці 3. Реалізація заснована на формулі:

$$d = \|x, y\| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \quad ,$$

де d – це відстань, x – координати першого користувача, y – координати другого користувача, n – загальна кількість користувачів, i – номер поточного користувача.

Таблиця 3

Результати знаходження Евклідової відстані

	Олександр		Валерія		Олена		Павло	
	ВР	Python	ВР	Python	ВР	Python	ВР	Python
Олександр	X		0,082	0,082	0,217	0,217	0,125	0,125
Валерія	0,082	0,082	X		0,072	0,072	0,24	0,24
Олена	0,217	0,217	0,072	0,072	X		0,125	0,125
Павло	0,125	0,125	0,24	0,24	0,125	0,125	X	

Кореляція Пірсона – це трохи складніший спосіб визначення подібності інтересів людей. Коефіцієнт кореляції – це показник того, наскільки добре два набори даних вписуються в пряму лінію [2]. Кореляція Пірсона обраховується за формулою:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad ,$$

де d – це відстань, x – координати першого користувача, y – координати другого користувача, n – кількість користувачів, i – поточний користувач.

Результат обчислень подібностей за допомогою власної реалізації та функції pearsonr з бібліотеки scipy.stats мови Python наведений у таблиці 4.

Таблиця 4

Результати знаходження кореляції Пірсона

	Олександр		Валерія		Олена		Павло	
	ВР	Python	ВР	Python	ВР	Python	ВР	Python
Олександр	X		1,678	0,5	1,867	0,5	0	NaN
Валерія	1,678	0,5	X		2,713	7,46	1,99	inf
Олена	1,867	0,5	2,713	7,46	X		0	NaN
Павло	0	NaN	1,99	inf	0	NaN	X	

Манхеттенська метрика розглядається X. Мінковським як спосіб математики, в якому звичайна функція подібності або евклідова відстань замінюється новою метрикою, в якій відстань між двома точками є сумою абсолютних відмінностей їхніх координат [3]. Обчислюється за формулою:

$$d = \|x, y\| = \sum_{i=1}^{n} |x_i - y_i| \ ,$$

де d – це відстань, x – координати першого користувача, y – координати другого користувача, n – загальна кількість користувачів, i – номер поточного користувача.

Результат роботи коду, що реалізує цей метод за допомогою власної реалізації та функції manhattan_distances з бібліотеки sklearn.metrics.pairwise мови Python, наведений у таблиці 5.

Таблиця 5

Результати знаходження Манхеттенської метрики

	Олександр		Валерія		Олена		Павло	
	ВР	Python	ВР	Python	ВР	Python	ВР	Python
Олександр	X		0,0625	0,0625	0,167	0,166	0,125	0,125
Валерія	0,0625	0,0625	X		0,04	0,043	0,2	0,2
Олена	0,167	0,166	0,04	0,043	X		0,125	0,125
Павло	0,125	0,125	0,2	0,2	0,125	0,125	X	

Косинусна відстань – це значення відстані між двома векторами внутрішньої площі, який вимірює косинус кута між ними. Також відома як векторна подібність [4]. В реалізації використовувалась формула:

$$d = \cos(\vec{i} \cdot \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|^2 \cdot \|\vec{i}\|^2} \quad ,$$

де d – відстань між користувачами, i – вектор першого користувача, j – вектор другого користувача.

Результат обчислень подібностей за допомогою власної реалізації та функції cosine_distances з бібліотеки sklearn.metrics.pairwise мови Python наведений у таблиці 6.

Таблиця 6

Результати знаходження косинусної відстані

	Олександр		Валерія		Олена		Павло	
	BP	Python	BP	Python	BP	Python	BP	Python
Олександр	X		0,501	0,501	0,508	0,508	0,5	0,5
Валерія	0,501	0,501	X		0,69	0,69	0,5023	0,5023
Олена	0,508	0,508	0,69	0,69	X		0,5	0,5
Павло	0,5	0,5	0,5023	0,5023	0,5	0,5	X	

Проаналізуємо кожен метод за однаковими критеріями:

1. Схожість результатів власної реалізації та методу з готових бібліотек мови Python. За цим критерієм не підходить лише кореляція Пірсона.
2. Наявність нестандартних (0, NaN та нескінченність) значень в результатах. За цим критерієм також програє кореляція Пірсона.
3. Логічність отриманих результатів. На початку ми визначили, що користувачі Олександр та Олена схожі у своїх вподобаннях, отже від цього і будемо відштовхуватись. Значення схожості, отримані різними алгоритмами, вказані в Таблиці 7.

Легко побачити, що Евклідова відстань та Манхеттенська метрика показали кращі результати, ніж Косинусна відстань. Тому для використання оберемо найпростіший алгоритм – метод знаходження Евклідової відстані.

Таблиця 7

Значення схожості користувачів

	Евклідова відстань	Манхетенська метрика	Косинусна відстань
Олександр - Олена	0,217129273	0,1666666667	0,5081080666
Олександр - Валерія	0,08209951522	0,0625	0,5015337568

Алгоритм колаборативної фільтрації.

В алгоритмі колаборативної фільтрації використовується інформація про поведінку користувачів в минулому – наприклад, інформація про покупки або оцінки [5]. У цьому методі рекомендації надаються з урахуванням поведінки усіх, чиї вподобання схожі на смаки даного користувача [5]. Отже, основна ідея колаборативної фільтрації – схожим користувачам зазвичай подобаються схожі курси.

Перейдемо до надання рекомендацій. У таблиці 8 показаний процес надання рекомендацій користувачу Олександр.

Таблиця 8

Демонстрація процесу надання рекомендацій

Користувач	**Схожість**	**Фронт-енд**	**S.x Фронт-енд**	**UI/UX**	**S.x UI/UX**
Валерія	0.008	6	0.048	7	0.056
Олена	0.071			1	0.071
Павло	0.02	9	0.18	8	0.16
Сума			*0.228*		*0.287*
Сума схожості			*0.028*		*0.099*
Результат			**8.14**		**2.9**

Тут показані кореляційні оцінки для кожного користувача та їхні рейтинги курсам (Фронт-енд та UI/UX), які не оцінив Олександр. Стовпці з «S.x», дають подібність, помножену на рейтинг. Рядок "Сума" показує суму всіх цих оцінок.

Алгоритм фільтрації заснований на вмісті.

Фільтрація на основі вмісту або тематична фільтрація також формує рекомендацію на основі поведінки користувача. Проте, цей підхід

використовує інформацію про технології та характеристики цих технологій. Тобто, алгоритм шукає користувачу курси, що схожі на вже пройдені ним. Як відзначалося раніше, пошук схожих курсів відбувається так само, як і пошук схожих користувачів, отже можна використати евклідову відстань. В таблиці 9 показане надання рекомендацій користувачу Олександр.

Таблиця 9

Рекомендації для користувача Олександр

Курс	**Оцінки**	**Фронт-енд**	**R.x Фронт-енд**	**UI/UX**	**R.x UI/UX**
Бази даних	13	0,25	3,25	0,085	0,61
Комп'ютерні мережі	10	0,143	1,43	0,167	0,52
Xamarin	6	0,167	1,002	0,091	0,546
Сума		0,56	5,682	0,343	1,676
Нормалізація			**10,14642857**		**4,88629738**

Значення в стовпці «Курс» позначають курси, що пройшов Олександр, в стовпці «Оцінки» - оцінки за курси з першого стовпця. В стовпцях «Фронт-енд» та «UI/UX» - значення схожості на курс зі стовпця «Курс». В стовпцях з префіксами R.x - значення, які приблизно показують передбачену оцінку Олександра за даний курс. Таблиця 9 схожа за структурою на таблицю 8.

Отже, можемо порівняти результати обох алгоритмів (таблиця 10).

Таблиця 10

Результати передбачення оцінок Користувачу Олександр

	Колаборативна фільтрація	Фільтрація на основі вмісту
Фронт-енд	8.14	10,14
UI/UX	2.9	4,88

Результати вийшли дуже схожими, отже було прийнято рішення, що обидва методи повинні використовуватись в реалізації системи дистанційного навчання.

Алгоритм фільтрації заснованої на вмісті буде надавати рекомендації новим користувачам, тому що у системи ще не буде даних про пройдені ними курси.

Алгоритм колаборативної фільтрації надаватиме рекомендації досвідченим користувачам, які вже взаємодіяли з системою, а отже їм можна рекомендувати щось на основі їхніх вподобань. Звичайно, досвідчені користувачі зможуть отримувати рекомендації також за допомогою алгоритму фільтрації заснованої на вмісті.

Висновки. Рекомендаційна система – це програма, яка на основі даних про користувача і елемент передбачає та надає релевантні рекомендації. Рекомендаційні системи можна використовувати й у дистанційному навчанні, де можна багато чого запропонувати: курси, категорії курсів, нові технології для опанування**.**

Було проаналізовано методи знаходження схожих елементів, а саме: Евклідова відстань, кореляція Пірсона, манхеттенська метрика та косинусна відстань. Найкращі результати показали Евклідова відстань та манхеттенська метрика, проте для використання була обрана Евклідова відстань.

Розглянуті два методи надання рекомендацій: алгоритм колаборативної фільтрації та алгоритм фільтрації заснованої на вмісті. В результаті тестувань було виявлено, що обидва методи надають релевантні рекомендації, отже було вирішено, що досвідчені користувачі системи отримуватимуть рекомендації за допомогою обох методів, а нові користувачі, внаслідок відсутності достатньої кількості даних, – лише за допомогою алгоритму фільтрації заснованої на вмісті.

Література

1. Як працюють рекоменаційні системи. Лекція в Яндексі [Електронний ресурс] / Хабр. – 2014. – Режим доступу до ресурсу: https://habr.com/company/yandex/blog/241455/ – Дата доступу: 22.02.2018.

2. Сегаран Т. Програмуємо колективний розум / Тобі Сегаран. – Себастопол: O'Reilly Media, 2007. – 360 с.
3. Манхеттенська метрика [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D0%BD%D1%85%D0%B5%D1%82%D1%82%D0%B5%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B0 – Дата доступу: 26.02.2018.
4. Раджараман А. Видобуток масивних наборів даних / Раджараман А., Уллман Д. – Кембридж: Cambridge Press, 2012 – С. 245.
5. Брейє Д. Емпіричний аналіз алгоритмів прогнозування колаборативної фільтрації. / Брейє Д., Хеккерма Д., Каді С., 1998 – С. 243.