DOI: https://doi.org/10.26565/2304-6201-2024-61-01

УДК (UDC) 004.08

Бакуменко Ніна Станіславівна *к.т.н., доцент, доцент кафедри теоретичної та прикладної системотехніки, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: n.bakumenko@karazin.ua;* **https://orcid.org/0000-0003-3496-7167**

Толстолузька Олена Геннадіївна *д.т.н., с.н.с., професор кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, м. Харків-22, Україна, 61022 e-mail: elena.tolstoluzka@karazin.ua;* **https://orcid.org/0000-0003-1241-7906**

Ясінський Ярослав Андрійович *студент; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: yar.yasinskyi@gmail.com;* **https://orcid.org/0009-0008-0460-5687**

Аналіз алгоритмів кластеризації для надання рекомендацій товарів

Актуальність. У сучасному світі, насиченому широким спектром товарів та послуг, питання надання персоналізованих рекомендацій для вибору стає актуальним завданням для багатьох сфер, зокрема електронної комерції та онлайн-платформ. Рекомендаційні системи, що працюють на основі пошукових алгоритмів та алгоритмів кластеризації, мають потенціал для значного покращення користувацького досвіду, пропонуючи релевантні та персоналізовані пропозиції товарів. Одними з ключових переваг використання алгоритмів кластеризації для рекомендаційних систем є можливість прогнозувати схожість елементів в залежності від відповідності до певної характеристики, завдяки чому можливо реалізувати ефективний пошук товарів за характеристиками. Внаслідок чого з'являється можливість є сегментувати базу користувачів на окремі підгрупи, що можуть представляти різні сегменти ринку, групи за вподобаннями, цільову аудиторію певних товарів. Виявлення проблем і недоліків таких систем дозволяє вдосконалювати алгоритми, що призводить до більш точних прогнозів і та збільшення продажів компаній.
Мета. Мета даної статті полягає в аналізі ефективності використання методів кластерного аналізу в задачах формування рекомендацій.
Методи дослідження. Порівняльний аналіз, експеримент.
Результати. Проведено аналіз ефективності алгоритмів кластеризації різних типів (k -means++, Mean Shift та HDBSCAN) для надання рекомендацій товарів на основі оцінювання відповідності запиту користувача у відсотковому відношенні, використання оперативної пам'яті, та час виконання запиту. Серед розглянутих найкращі характеристики показав алгоритм k-means++.
Висновки. Проведений аналіз підтверджує ефективність використання методів кластерного аналізу в рекомендаційних системах. Виявлення проблем і недоліків таких систем дозволяє вдосконалювати алгоритми, що призводить до більш точних прогнозів і та збільшення продажів компаній.

Ключові слова: *алгоритм кластеризації, рекомендаційна система, k-means++, HDBSCAN, Mean Shift.*

Як цитувати: Бакуменко Н. С., Толстолузька О. Г., Ясінський Я. А. Аналіз алгоритмів кластеризації для надання рекомендацій товарів. *Вісник Харківського національного університету імені В.Н.Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2024. вип. 61. С.6-13. https://doi.org/10.26565/2304-6201-2024-61-01

How to quote: N.S. Bakumenko, O. G. Tolstoluzka, Y. A. Yasinskyi, "Analysis of clustering algorithms for providing product recommendations", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modeling. Information Technology. Automated control systems,* vol. 61, pp.6-13, 2024. [In Ukrainian]. https://doi.org/10.26565/2304-6201-2024-61-01

1 Вступ

В сучасному світі електронної комерції успішне надання рекомендацій товарів є важливою складовою для підвищення якості торгівельних сервісів, ефективності реклами та збільшення прибутків компаній. Ефективна система рекомендацій може сприяти підвищенню конкурентоспроможності платформи, збільшуючи продажі завдяки персоналізованим пропозиціям. Системи рекомендацій, що пропонують товари відповідно до індивідуальних уподобань користувачів, значно покращують досвід покупців, стимулюючи їх до повторних покупок. Одним із методів, який використовується в таких системах, є кластеризація даних.

Кластеризація – це процес розподілу набору об'єктів на групи (кластери) таким чином, щоб об'єкти в одному кластері були більш схожі один на одного, ніж на об'єкти в інших кластерах. Застосування алгоритмів кластеризації в системах рекомендацій дозволяє групувати користувачів або товари на основі їх характеристик та поведінки, що в свою чергу сприяє наданню більш релевантних рекомендацій.

У даній роботі статті досліджується ефективність різних алгоритмів кластеризації для надання рекомендацій товарів. Основною метою є аналіз та порівняння цих алгоритмів з точки зору точності рекомендацій та їх відповідності очікуванням користувачів. Для досягнення цієї мети було створено спеціальний набір даних з реальної бази існуючих товарів, реалізовано методи надання рекомендацій на основі кластеризації, та проведено оцінка ефективності роботи цих методів.

2 Використання алгоритмів кластеризації для надання рекомендацій

Кластеризація є методом, який часто використовується для рекомендаційних систем, оскільки вона дозволяє ідентифікувати групи користувачів зі схожими смаками. Це сприяє більш цілеспрямованим рекомендаціям, використовуючи переваги отримання інформації вже визначених вподобань клієнтів [1, 2]. В роботі [3] було показано, що кластеризація може допомогти подолати такі проблеми, як розрідженість даних і масштабованість, які часто виникають при формуванні рекомендацій, і, таким чином, сприяти побудові більш точних і різноманітних рекомендацій.

2.1 Кластеризація на основі центроїдів

Розглянемо групу алгоритмів кластеризації, робота яких заснована на визначенні центрів кластерів. Принцип роботи алгоритму побудований на використанні групування подібних точок властивостей товарів у кластери шляхом встановлення репрезентативних точок, які є центроїдами. Відомим алгоритмом на основі центроїда є k-means, який працює шляхом виділення подібних точок даних, наприклад, користувачів або елементів у кластери, що представлені центроїдами. Головним недоліком цього алгоритму у персоналізованих системах рекомендацій є чутливість до стартового вибору центроїдів. Якщо початкові центроїди не вибрані ретельно, алгоритм може сходитися до локального оптимуму, що призведе до неоптимальної кластеризації [4, 5].

Існує покращена версія алгоритму, що має назву k-means++ – це варіант k-means, що усуває його основний недолік, тобто чутливість до початкових умов. Центр першого кластеру обрається випадково, а центр наступного кластера обирається як найбільш віддалений від попереднього. Це усуває основний недолік алгоритму, пов’язаний з чутливістю до вибору початкових даних [6].

2.2 Ієрархічна кластеризація

Ієрархічна кластеризація базується на підході аналізу елементів, що більш пов’язані з елементами поряд, а ніж з тими, що розташовані далі. Цей тип алгоритмів створює деревоподібні структури з метою генерації кластерів. Основним підходом для створення кластера є підбір елементів зі структури, за відстанню, тоді створений фрагмент даних характеризується максимальною відстанню, для групування частин кластера.

Цікавим прикладом є алгоритм HDBSCAN, що ієрархічним алгоритм кластеризації, який особливо підтвердив свою ефективність у використанні для персоналізованих систем рекомендацій. Цей алгоритм кластеризації базується на щільності розподілу, що означає, що він групує точки датасету разом на основі їх щільності в просторі даних. Це робить HDBSCAN більш стійким до викидів і шуму, ніж інші алгоритми кластеризації [7]. Додатковою перевагою визначення кількості кластерів в процесі роботи алгоритму, н відміну від k-means, і можливість ідентифікувати кластери різної форми.

Задля використання HDBSCAN для персоналізованих систем рекомендацій, першим кроком є об'єднання користувачів у групи зі схожими перевагами, що можна зробити шляхом кластеризації користувачів на основі їхніх попередніх оцінок елементів, таких як фільми, книги чи продукти. Після об'єднання користувачів у кластери наступним кроком є створення персоналізованих рекомендацій для кожного користувача. Це можна зробити, порекомендувавши елементи, популярні серед користувачів у визначеному кластері.

2.3 Кластеризація на основі щільності

Кластеризація на базі щільності використовує ідею ідентифікації груп елементів в даних через припущення, що кластер у просторі даних є безперервною областю високої щільності, відокремленої від інших таких кластерів суміжними областями низької точки щільності. Точки даних у роздільних областях із низькою щільністю точок зазвичай вважаються шумом чи викидами.

Розглянемо цей тип кластеризації на прикладі алгоритму Mean shift. Перший крок передбачає оцінку базової функції щільності ймовірності розподілу точок даних. Зазвичай це робиться за допомогою оцінки щільності ядра, де кожна точка даних представлена функцією ядра з центром у цій точці. Функція ядра визначає вагу, призначену кожній точці даних у процесі оцінки щільності. Наступним кроком алгоритм ітеративно переміщує точки даних до областей з вищою щільністю на основі векторів середнього зсуву, обчислених як зважене середнє значення відмінностей між кожною точкою та її сусідами. Ітерації тривають до конвергенції, що вказується векторами мінімального середнього зсуву. Кінцеве розташування кожної точки даних позначає центр кластера, що дозволяє призначити найближчий центр і ідентифікувати окремі кластери в даних [8].

На відміну від добре відомого підходу кластеризації k-means, Mean shift не потребує припущень щодо кількості кластерів і форми розподілу, але його продуктивність залежить від вибору параметрів масштабу. Пропускна здатність є єдиним параметром для налаштування, тому для одновимірного випадку це відносно проста процедура, але в багатовимірному випадку це може викликати певні складнощі [9].

Для розгляду в контексті використання в рекомендаційних системах були обрані три алгоритми, такі як k-means++, HDBSCAN та Mean Shift, кожен з яких відноситься різних типів кластеризації.

3 Тестовий набір даних

Для тестування роботи алгоритмів було створено датасет з реальними даними про мікрохвильові печі з українських відкритих джерел, з переліком характеристик цифрових та категоріальних. Для тестування надання персоналізованих рекомендацій було обрано об'єм датасету розміром 100 елементів. Дані для датасету про існуючі товари з отримані з агрегаторів товарів у листопаді 2023 року.

Створений датасет містить 9 типів характеристик, що описують товар, окрім id_product, що є ідентифікаційним номером, та не використовується в обчисленнях. Категоріальними даними є колір, виробник та назва виробу. Характеристики ширини, глибини та висоти, містять значення з плаваючою точкою (рис. 1). Ціна, потужність, та максимальна споживана потужність – цілі числа.

	A	B	C	D	E	F	G	H	I	J
1	id_product	manufacturer	name	cost	color	width	height	depth	power	max_consumption
2	1	Gorenje	MO17E1B	2689	black	45.5	26.1	35.3	700	1150
3	2	ERGO	Y35MW	2313	white	44.6	24.3	33.8	700	1100
4	3	Edler	ED-2067W	1961	white	44.4	24.1	35.8	700	1150
5	4	MILANO	MW-4001W	1999	white	45	25	37	700	1100
6	5	Grunhelm	20MX701-W	1899	white	44.8	24.5	32.9	700	1000
7	6	ERGO	EM-2040	2070	black	44	25.9	34.3	700	1050
8	7	Edler	ED-2079W	2059	white	45.1	25.9	33.5	700	1150
9	8	Mirta	Elegance MW-2510W	1900	white	44.6	24.3	33.2	800	1100
10	9	Hansa	AMGF17M2BH	2299	black	45.2	26.2	31.5	600	1000
11	10	Liberton	LMW-2077M	2165	black	44	35.5	25.9	700	1100

Рисунок 1. Приклад перших 10 записів сформованого набору даних

4 Опис обраних технологій

Програмна реалізація була виконана на мові Python з використанням функцій бібліотеки Scikit-learn. Пакет надає набір інструментів для таких завдань, як розробка комп'ютерних моделей, калькуляція метрик для оцінювання ефективності, а також різноманітні доповнення, що наприклад включають функції попередньої обробки інформації датасету [10]. Також було використано бібліотеку Pandas, яка надає інтсрументи для різноманітної маніпуляції даними, зокрема імпорту даних з баз даних, електронних таблиць, файлів формату CSV, та інших [11].

Для оцінки використання пам'яті програмою був використаний профілізатор пам'яті, основне призначення якого виявлення витоків пам'яті та покращення використання пам'яті у програмах на Python. Профілізатор пам'яті аналізує ефективність використання місця в коді та характеристики використовуваних пакетів, пропонуючи ідеї для оптимізації використання пам'яті. Пакет memory_profiler перевіряє використання пам'яті інтерпретатором на кожному рядку. Стовпчик інкрементів дозволяє виявити місця в коді, де виділяються великі обсяги пам'яті. Це особливо важливо при роботі з масивами [12].

Розрахунки проводилися на базі процесору Intel(R) Core(TM) i7-7700HQ CPU, з частотою 2,80 ГГц. Вбудована оперативна пам'ять - 16,0 ГБ.

5 Оцінка ефективності алгоритмів кластеризації у системі рекомендацій

Для реалізації системи рекомендацій на базі кластеризації необхідно дослідити ефективність роботи моделі на різних алгоритмах кластеризації. Для оцінки ефективності надання рекомендацій були використані три параметри: відповідність запиту користувача у відсотковому відношенні, використання оперативної пам'яті, та час виконання запиту.

Перший кроком проводилася кластеризація на основі введеного масиву характеристик. Як результат, були отримані рекомендації на основі відповідного запиту кластеру.

Слід зазначити, запит від користувача можна розглядати як ідеальний товар, що йому необхідний. Набір даних може не містити обраного елементу. Тоді задачею моделі пошук елементів найбільш подібних обраному, та розрахунок цієї подібності.

Після цього відбувається перевірка вимогам користувача за допомогою розрахунку евклідової відстані для обраного товару та кожного елементу у цьому списку [13].

Для коректної роботи алгоритму k-means++ необхідно було визначити кількість кластерів. Для цього можна використати метод ліктя, що визначає оптимальну кількість кластерів у наборі даних шляхом побудови залежності деякої цільової функції якості кластеризації від кількості кластерів.

На рис. 2 зображено графік залежності внутрішньокластерної дисперсії від кількості кластерів для створеного набору даних. «Точка ліктя» розташована там, де швидкість зниження різко змінюється, та вказує на оптимальну кількість кластерів для використання в алгоритмі кластеризації [14], у даному випадку – визначена кількість кластерів дорівнює 3.

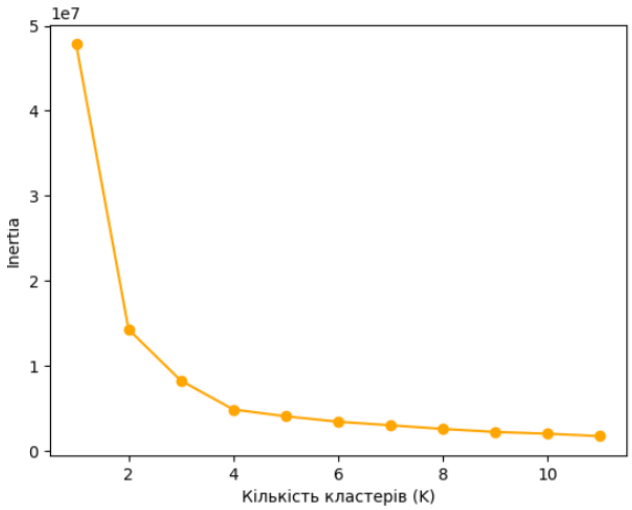

Рисунок 2. Графік залежності внутрішньокластерної дисперсії від кількості кластерів

Для оцінки точності роботи моделі, нам необхідно ввести характеристики наявного товару з набору даних та перевірити, як надаються рекомендації. Якщо евклідова відстань введеного

товару близька до 0, то можна констатувати 100 відсотковий збіг – отже, модель працює правильно. Для порівняння результатів різних алгоритмів необхідно порівняти відстані схожих елементів, що відрізняються за однією чи кількома характеристиками. До прикладу, оберемо першу мікрохвильову піч з характеристиками зазначеними на рис. 1.

Перший продукт у таблиці (M017E1B) має показник евклідової відстані 0,00. Це тому, що продукт ідентичний сам собі. Всі алгоритми змогли визначити наявність цього елементу в наборі даних, отже створена реалізація працює коректно. Інші елементи в таблиці мають оцінку схожості, що перевищує 0,00. Це означає, що продукт схожий на M017E1B, та може бути запропонований користувачу.

Після запуску моделі на основі алгоритму k-means++ модель може визначити який елемент був введений через те, що модель виставила 100 відсотків збігу введеному елементу. Додатково отримано список десяти рекомендацій товарів (рис. 3).

Другий продукт (M017E1W) є схожим на товар, на основі якого виконуються рекомендації, але не таким самим, оскільки в нього відмінне одне значення – колір, тоді різниця в один категоріальний параметр складає 10.07.

Останній елемент у списку рекомендацій k-means ++ (PMW 20711 KB) має значення – 331.45**,** отже має схожість з обраним товаром, на основі якого робляться рекомендації, але може бути значно відмінним.

Результати ієрархічної кластеризації гірші аніж двох методів перед ним. Найбільша відстань складає 796.86.

Рекомендації K-means++			Рекомендації Mean Shift			Рекомендації HDBSCAN		
Id	Product	Similarity	Id	Product	Similarity	Id	Product	Similarity
0	M017E1B	0.00	0	M017E1B	0.00	0	M017E1B	0.00
18	M017E1W	10.07	18	M017E1W	10.07	18	M017E1W	10.07
19	M020E1WH	130.34	13	PMW 20757 HB	100.35	17	PMW 20711 KW	126.33
22	M017E1S	152.00	17	PMW 20711 KW	126.33	19	M020E1WH	130.34
11	MW-4001BR	199.42	19	M020E1WH	130.34	11	MW-4001BR	199.42
20	LMW-2074M	211.43	11	MW-4001BR	199.42	24	MW-MM-20P(WH)	220.47
28	M017E1BH	251.16	14	PMW 20711 KB	331.45	34	AMG20M70GSVH	398.25
29	LMW-2079M	284.72	1	Y35MW	379.32	8	AMGF17M2BH	429.67
31	R200BKW	322.38	15	PMW 20715 KB	379.42	45	MW-4010B	612.05
14	PMW 20711 KB	331.45	12	PMW 20757 HW	382.12	7	Elegance MW-2510W	796.89

Рисунок 3. Списки рекомендованих товарів створених за допомогою різних алгоритмів кластеризації

Для визначення кращого методу для надання персоналізованих рекомендацій було обрано збіжні елементи з кількох списків, отриманих із застосуванням різних алгоритмів кластеризації.

Для порівняння ефективності надання рекомендацій запропонованих методів підраховано середню суму відстаней, максимальну відстань та підсумкову суму відстаней для усіх кластерів.

Результати розрахунків задані у табл. 1:

Таблиця 1. Порівняння результатів надання рекомендацій

Евклід. відстань	K-means++	Mean Shift	HDBSCAN
Максимальна	331.45	382.12	796.89
Середня	210.33	237.5	324.83
Сума	1892.97	2137.57	2923.49

Таблиця містить порівняльний аналіз трьох алгоритмів кластеризації – k-means++, Mean Shift та HDBSCANз використанням евклідової відстані для оцінки точності рекомендацій товарів. Евклідова відстань вимірює схожість між рекомендованими товарами та запитаним товаром.

Для тестування часу було вирішено провести декілька ітерацій запуску моделі, щоб переконатися, що результат не був випадковим. Під час кожної спроби вимірювався час, необхідний для виконання моделі кластеризації на тестовому наборі даних. Підсумковим результатом є середнє значення часу виконання для кожного алгоритму. Час роботи алгоритму наведений у табл. 2:

Таблиця 2. Час роботи алгоритму (сек.)

Спроба	K-means++	Mean Shift	HDBSCAN

1	0.86	1.83	0.32
2	0.73	1.79	0.29
3	0.74	1.79	0.33
4	0.73	1.80	0.32
5	0.79	1.81	0.31
Сер. значення	0.77	1.804	0.314

HDBSCAN демонструє найшвидший час виконання серед трьох алгоритмів кластеризації. Середній час виконання складає лише 0.314 секунди, що робить його найбільш ефективним за цим параметром.

K-means++ займає друге місце за швидкістю виконання із середнім часом 0.77 секунди. Хоча він працює повільніше за HDBSCAN, все ж таки час його виконання залишається досить низьким. Результати між декількома запусками одного алгоритму відрізняються на незначну частину, однак найбільша різниця між ітераціями складає 0.09 сек, що може бути пов'язано з першим запуском моделі.

Mean Shift є найповільнішим алгоритмом серед розглянутих, зі середнім часом виконання 1.804 секунди. Це свідчить про його менш ефективну роботу з точки зору часу.

Порівняння обсягу оперативної пам’яті для різних алгоритмів кластеризації наведено в таблиці 3.

Таблиця 3. Використання оперативної пам'яті (MiB)

Спроба	K-means++	Mean Shift	HDBSCAN
1	1.4	1.5	0.8
2	1.4	1.5	0.9
3	1.5	1.6	0.8
4	1.4	1.6	0.9
5	1.4	1.5	0.8
Сер. значення	1.42	1.54	0.84

За результатами експериментів проведених за допомогою Memory profiler за вимірами обсягу пам'яті можна сказати, що найменше споживає пам'яті ієрархічна кластеризація. HDBSCAN демонструє найменше використання оперативної пам'яті серед трьох алгоритмів кластеризації. Середнє використання пам'яті складає лише 0.84 MiB, що робить його найбільш ефективним з точки зору економії пам'яті. Кластеризація на основі щільності показує найбільші результати по споживанню пам'яті – 1.54 MiB, що може пояснюватися тим, що даний підхід, який зазвичай використовується для розпізнавання зображень. K-means++ займає друге місце за обсягом використання пам'яті із середнім значенням 1.42 MiB. Кластеризація на основі центроїдів вимагає трохи менше пам'яті, проте все одно більше ніж HDBSCAN.

6 Висновки

У даному дослідженні проведено аналіз ефективності різних алгоритмів кластеризації для надання рекомендацій товарів. Було розглянуто три алгоритми: k-means++, Mean Shift та HDBSCAN. Для дослідження було створено спеціальний датасет з характеристиками товарів, на основі якого проводилося тестування.

Результати дослідження показали, що алгоритм k-means ++ надає найбільш точні та відповідні рекомендації порівняно з іншими обраними алгоритмами. За результатами тестування k-means ++ демонструє найменшу середню евклідову відстань між рекомендованими товарами та обраним товаром, що вказує на високу точність рекомендацій. Зокрема, максимальна евклідова відстань для K-means++ склала 331.45, що є нижчим показником у порівнянні з Mean Shift (382.12) та HDBSCAN (796.89).

У той же час, алгоритм HDBSCAN показав найкращі результати за часом виконання запитів, що може бути корисним у випадках, коли швидкість є критично важливою. Середній час виконання для HDBSCAN складав лише 0.314 секунд, тоді як для K-means++ та Mean Shift цей показник був значно вищим.

За обсягом використання оперативної пам'яті алгоритм HDBSCAN також показав найкращі результати, споживаючи в середньому 0.84 MiB, що значно менше ніж Mean Shift (1.54 MiB) та K-means++ (1.42 MiB).

Таким чином, вибір алгоритму кластеризації для системи надання рекомендацій товарів залежить від конкретних вимог до системи. Якщо пріоритетом є точність рекомендацій, K-means++ є найкращим вибором.

Отже, результати дослідження підтверджують доцільність використання кластеризації для покращення систем рекомендацій товарів та дозволяють зробити обґрунтований вибір алгоритму залежно від специфічних вимог до системи.

СПИСОК ЛІТЕРАТУРИ

1. How Search Engine Personalization Affects Rankings. [Online]. Available: https://marketbrew.ai/how-search-engine-personalization-affects-rankings Accessed on: May 21, 2024.
2. Data Clustering: Intro, Methods, Applications. [Online]. Available: https://encord.com/blog/data-clustering-intro-methods-applications Accessed on: May 22, 2024.
3. J. Das, S. Majumder, K. Mali, “Clustering Techniques to Improve Scalability and Accuracy of Recommender Systems*”, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 29, no. 04, pp.. 621–651, 2021
4. k-means Advantages and Disadvantages: [Online]. Available: https://developers.google.com/machine-learning/clustering/ Accessed on: May 22, 2024.
5. Artley B. Unsupervised Learning: k-means Clustering. Towards Data Science: [Online]. Available: https://towardsdatascience.com/unsupervised-learning-k-means-clustering-27416b95af27 Accessed on: May 20, 2024.
6. D. Arthur,S. Vassilvitskii, "k-means++: the advantages of careful seeding", *in Proc. of the Eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia, PA, USA., 2007, pp. 1027–1035.
7. Christopher A. Hierarchical Clustering and Density-Based Spatial Clustering of Applications with Noise (DBSCAN): Online]. Available: https://medium.com/mlearning-ai/hierarchical-clustering-and-density-based-spatial-clustering-of-applications-with-noise-dbscan-b8d903095532 Accessed on: May 10, 2024.
8. J. Sander, "Density-Based Clustering", in *Encyclopedia of Machine Learning*,. C. Sammut, G. I. Webb, Eds. Boston, MA, USA:Springer, 2011, pp. 349-353.
9. Damir Demirović, "An Implementation of the Mean Shift Algorithm", *Image Processing On Line*, no. 9, pp. 251–268, 2019.
10. Scikit-learn User Guide: [Online]. Available: https://scikit-learn.org/stable/user_guide.html Accessed on: May 22, 2024.
11. Pandas documentation: [Online]. Available: https://pandas.pydata.org/ Accessed on: May 24, 2024.
12. Memory-profiler: [Online]. Available: https://pypi.org/project/memory-profiler/ Accessed on: May 24, 2024.
13. Euclidean distance score and similarity. Available: https://stats.stackexchange.com/questions/53068/euclidean-distance-score-and-similarity Accessed on: May 24, 2024.
14. Elbow Method for optimal value of k in k-means? Available: https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/ Accessed on: May 24, 2024.

Bakumenko Nina

Candidate of Technical Sciences; Associate Professor of theoretical and applied system engineering department;
V.N. Karazin Kharkiv National University
Svobody Sq 4, Kharkiv, Ukraine, 61022
e-mail: n.bakumenko@karazin.ua;
https://orcid.org/0000-0003-3496-7167

Tolstoluzka Olena *doctor of Technical Sciences; Professor of theoretical and applied system engineering department;*
V.N. Karazin Kharkiv National University
Svobody Sq 4, Kharkiv, Ukraine, 61022
e-mail: elena.tolstoluzka@karazin.ua;
https://orcid.org/0000-0003-1241-7906

Yasinskyi Yaroslav *student;*
V.N. Karazin Kharkiv National University
Svobody Sq 4, Kharkiv, Ukraine, 61022
e-mail: yar.yasinskyi@gmail.com;
https://orcid.org/0009-0008-0460-5687

Analysis of clustering algorithms for product recommendations

Relevance. In today's world, where a wide range of goods and services are available, the task of providing personalized recommendations for selecting the right one is becoming an increasingly important in many areas, including e-commerce and online platforms. Expert recommendation systems powered by search and clustering algorithms have the potential to significantly improve the user experience by offering relevant and personalized product suggestions. One of the key advantages of using clustering algorithms for recommender systems is the ability to predict the similarity of objects based on their compliance with a certain characteristic, which makes it possible to implement an effective search for products by characteristics. As a result, it allows dividing an user base into separate subgroups that can represent different market segments, preference groups, and the target audience of certain products. Identification of problems and shortcomings of such systems helps to improve algorithms, which leads to more accurate forecasts and increased sales.
Objective. The purpose of this article is to analyze the effectiveness of using cluster analysis methods in the tasks of generating recommendations.
Research methods. Comparative analysis, experiment.
Results. The effectiveness of clustering algorithms of different types (k-means++, Mean Shift and HDBSCAN) for providing product recommendations based on the assessment of the percentage of compliance with the user's request, the use of RAM, and the query execution time has been analyzed. The k-means++ algorithm showed the best performance among the tested algorithms.
Conclusions. Our analysis confirms the effectiveness of using cluster analysis methods in recommender systems. Identification of problems and shortcomings of such systems allows improving algorithms, which leads to more accurate forecasts and increased sales of companies.

Keywords: *clustering algorithm, recommender system, k-means++, HDBSCAN, Mean Shift.*

Надійшла у першій редакції 18.02.2024, в останній - 21.04.2024.
The first version has been received on 18.02.2024, the final version - on 21.04.2024.