

Аналіз існуючих алгоритмів музичних рекомендаційних систем

Ми живемо в час, коли людина просто переповнена інформацією. І, коли людина заходить на сайт пошуку музики, якогось товару чи відео, найменше, що вона хоче, – це поглибитись у ці вебресурси на довгий час, щоб знайти те, що їй потрібно. В таких ситуаціях на допомогу приходять рекомендаційні системи. Рекомендаційна система надає персоналізований список тих елементів, що мають сподобатися користувачу найбільше. Однією із найпопулярніших сфер, де застосовується рекомендаційна система, є музична. Заходячи на будь-який музичний стрімінговий сервіс (наприклад, Spotify), одразу бачиш список рекомендованих пісень, що можуть тобі сподобатися. Такі рекомендаційні системи розповсюджені по всьому інтернету і дозволяють зекономити нам багато часу та нервів при пошуку необхідного матеріалу. Під такими рекомендаційними системами можуть використовуватися різні алгоритми, які можна поділити на три великі групи: колаборативна фільтрація, фільтрація за вмістом та гібридна фільтрація. Кожен алгоритм має свої особливості і випадки використання, що розглядається більш детально в описі цих методів.

Також після детального аналізу було спроектовано власну музичну рекомендаційну систему. Було створено власний алгоритм побудови музичної рекомендаційної системи, оснований на попередньому аналізі вже існуючих алгоритмів. Було створено діаграму класів, в якій знаходяться необхідні сутності для реалізації музичної рекомендаційної системи на основі вподобань користувача.

Ключові слова: музична рекомендаційна система; колаборативна фільтрація; фільтрація за вмістом; гібридна фільтрація; колаборативна фільтрація за моделлю; колаборативна фільтрація за сусідством; алгоритм; рекомендація.

Актуальність теми. Музичний ринок з кожним роком продовжує зростати. Спираючись на дані International Federation of the Phonographic Industry (IFPI) – компанії, що займається аналітикою музичної індустрії, музичний ринок за останній рік збільшився на 18,5 % [1]. Так само зі збільшенням музичного ринку продовжують зростати музичні інтернет-платформи, найпопулярнішими з яких є стрімінгові сервіси. На цих платформах для прослуховування музики наявна велика кількість музичних треків, що може спантеличити користувача при пошуку музичного матеріалу, який йому потрібен. Для того щоб полегшити цей процес, доцільно створити системи, які б рекомендували матеріал користувачеві залежно від його вподобань. Такі системи називаються рекомендаційними.

Зі зростанням таких сервісів, як: Spotify, YouTube, Netflix тощо, рекомендаційні системи почали відігравати все більшу і більшу роль у нашому житті. Враховуючи також онлайн-рекламу, рекомендаційних систем вже неможливо уникнути в нашому повсякденні. Використання рекомендаційних систем поширено серед великої кількості додатків з різних областей, таких як: музика, книги, фільми, e-commerce, новини, телевізійні програми, веб, туризм тощо [2]. Через це є доцільним провести аналіз існуючих алгоритмів для проектування музичних рекомендаційних систем та, з врахуванням переваг та недоліків кожного алгоритму, створити власну рекомендаційну музичну систему, що буде надавати користувачеві персоналізовані рекомендації у різних додатках.

Розробка рекомендаційних систем, що б передбачали, що подобається користувачу максимально точно, становить інтересом досліджень не тільки наукових, але й бізнес-спільнот. Рекомендаційна система допомагає користувачеві обрати те, що йому потрібно; але в один і той самий час рекомендаційні системи допомагають і бізнесу отримати вигоду, якщо вони найбільш точно передбачать ту продукцію, яка найбільше сподобається користувачу. Наприклад, Netflix оголосив свій конкурс («Netflix prize») з винагородою у розмірі одного мільйона доларів, метою якого було створення такого алгоритму рекомендаційної системи, який би працював більш точно за той, що працював у компанії Netflix [3]. У статті в бізнес-журналі Школи Уортона Університету Пенсільванії заявляється, що рекомендаційні алгоритми Netflix показують користувачеві нішові продукти, які вже застаріли для того, щоб заохотити покупця платформи переглядати такі фільми. Великий інтерес у користувача значно зменшить інвентаризаційні затрати компанії, аніж якби Netflix робив замовлення на популярні блокбастери. Про це також дізналися такі компанії-гіганти, як Amazon та Apple [4].

Аналіз останніх досліджень та публікацій, на які спираються автори. Вичерпний аналіз алгоритмів популярних рекомендаційних систем було представлено в статті Емануїла Возаліса та

Константіноса Г. Маргарітіса [5]. Також вони запропонували свої методи покращення базового алгоритму. У [6] Сатія Пракаш Саху, Ананд Наутіял та Махенда Прасад «Machine Learning Algorithms for Recommender System – a comparative analysis» було експериментально проаналізовано існуючі алгоритми рекомендаційних систем на даних з ресурсу MovieLens. Результатом цього експерименту стало визначення, що найбільш ефективним алгоритмом є наївний баєсів класифікатор. Адіянсджа, Александр А. С. Гунаван та Дервін Сухартоно в роботі [7] описали створення своєї музичної рекомендаційної системи та дійшли до таких висновків: музична рекомендаційна система має враховувати інформацію про музичні жанри для отримання більш точних результатів та CRNN (Convolutional recurrent neural network), які враховують частотні ознаки та патерни часових послідовностей, мають кращу ефективність. Також детальний аналіз найпоширеніших алгоритмів рекомендаційних систем провели Прем Мелвіль та Вікас Сіндгвані [8]. Автори дійшли до висновку, що контент-орієнтовані методи в більшості випадків кращі за колаборативну фільтрацію, окрім деяких специфічних ситуацій. По-перше, колаборативна фільтрація працює краще у випадках, коли немає багато контенту, який асоціюється з елементами; де контент являється більш абстрактним (думки, ідеї тощо), що є важким для аналізу комп'ютером. По-друге, колаборативна фільтрація має властивість надавати рекомендації, які потрібні користувачу, але які не мають вмісту з профілю користувача.

Метою статті є аналіз існуючих алгоритмів для проектування та побудови рекомендаційних систем, вибір найкращого алгоритму та врахування переваг кожного з алгоритмів, що розглядаються для подальшого проектування власної музичної рекомендаційної системи.

Викладення основного матеріалу. Поняття «рекомендаційної системи» видозмінювалося за останні десятиліття років. Ось, наприклад, у статті Резніка та Варіана [9], що була написана у 1997 р., рекомендаційна система описується таким чином: «У звичайній рекомендаційній системі люди надають рекомендації як вхідні дані, які система агрегує і направляє до відповідних отримувачів. У деяких випадках головна трансформація відбувається в агрегації; в інших – цінність системи полягає в можливості здійснити правильну відповідність між рекомендувачем та тим, кому потрібна рекомендація».

Далі визначення розширилося з метою включити системи, які роблять рекомендації незалежно від того, як вони створені: «Будь-яка система, що створює індивідуальні рекомендації як вихідні дані чи має ефект допомоги користувачу персоналізовано з рекомендацією об'єктів у великому обсягу можливих варіантів» [10]. Більш загальне визначення було представлено Адомавічусом та Тужилінім: «Нехай C – множина всіх користувачів, а S – множина можливих елементів, які можуть бути рекомендовані. Нехай u – функція корисності, яка визначає корисність предмета s для користувача c : $C \times S \Rightarrow R$, де R – повністю впорядкований список (наприклад, беззнакові цілі або дійсні числа в певному проміжку). Тоді для кожного користувача $c \in C$, нам необхідно обрати елемент $s' \in S$, що максимізує корисність користувача» [11].

Що стало зрозуміло з попередніх визначень: в рекомендаційних системах фігурують дві сутності: *користувач* та *елемент*. Користувач – це той, хто використовує рекомендаційну систему і надає їй свої думки щодо різних предметів (елементів), а рекомендаційна система надає, зі своєї сторони, нові елементи користувачеві.

Що буде *вхідними даними* для рекомендаційної системи залежить від обраного алгоритму. Взагалі вхідними даними може бути щось, що належить до таких категорій:

- *оцінювання*, що передають думку користувача про якийсь предмет. Оцінки можуть бути представлені у різних формах. Наприклад, на багатьох музичних сайтах наявна система оцінювання від 0 до 100, де 0 означає найгіршу оцінку, а 100 – найкращу. Також популярною є п'ятизіркова система оцінювання від 0 зірок (найгіршої оцінки) до 5 зірок (найкращої оцінки). Поширеною практикою є бінарна система оцінювання від 0 до 1. Оцінки також можуть бути отримані з історії покупок користувача, з відвідування посилань та з інших видів шаблонів доступу до інформації;
- *демографічні дані*, що належать до таких даних, як: вік, стать, освіта користувачів. Ці дані складно отримати. Ця інформація отримується безпосередньо від користувача;
- *контент*, дані, що базуються на текстовому аналізі документів, пов'язаних з предметами, що оцінював користувач. Ознаки, що будуть отримані з аналізу, використовуються як вхідні дані для фільтраційного алгоритму з метою визначити профіль користувача.

Метою рекомендаційної системи є згенерувати пропозиції по новим предметам чи спрогнозувати корисність певного елемента для користувача.

Вихідними даними рекомендаційної системи може бути *передбачення* або *рекомендація*:

- *передбачення* визначається числовим значенням, яке являє собою очікувану думку користувача про певний предмет. Це передбачення має бути в тому самому числовому діапазоні, що і вхідні дані. Ця форма рекомендаційної системи відома як *індивідуальний рахунок*;
- *рекомендацією* вважається список з n елементів, що мають сподобатися користувачу найбільше. Вимогою цього списку являється те, що список повинен мати в собі тільки ті елементи, що користувач ще не придбав, прослухав, прочитав, подивився тощо. Ця форма рекомендаційної системи також відома під назвою *Топ-N рекомендацій*, або *Ранговий рахунок*.

Матриця оцінок користувача. Найбільш загальний вигляд в якому рекомендаційні системи вивчаються, зображений на таблиці 1.

Таблиця 1

Вигляд матриці оцінок користувача

		Елементи					
		1	2	...	i	...	m
Користувачі	1	91		23	56		
	2	41		72			18
	...		43			21	
	u			33	42		
	...		54				46
	n	64			100		
	a	71	?		55	96	

Відомі вподобання користувача показані матрицею з t користувачів та m предметів, де кожна клітинка $r_{u,i}$ відповідає оцінці, наданій предмету i користувачем u . Матриця оцінок користувачів у звичайних випадках являється розрідженою, тому що користувачі в більшості випадків не оцінюють більшість елементів. Задача рекомендаційної системи – спрогнозувати, яку оцінку надасть користувач попередньо неоціненному елементу. Зазвичай рейтинги прогноуються для елементів, які не переглядав користувач і найбільш оцінені предмети з'являються як рекомендації користувачеві.

Види алгоритмів фільтрації. Різні алгоритми фільтрації рекомендаційних систем можна категоризувати в 3 типи (рис. 1):

- *колаборативна фільтрація.* В системах, побудованих на цьому алгоритмі, користувачеві рекомендуються предмети, які основані на оцінюванні всіх користувачів разом;
- *фільтрація за вмістом.* Цей підхід визначається тим, що користувачеві рекомендуються елементи, що за вмістом схожі на предмети, що користувачеві колись подобались;
- *гібридна фільтрація.* Цей алгоритм, як можна здогадатись, об'єднує в собі колаборативну фільтрацію і фільтрацію за вмістом.

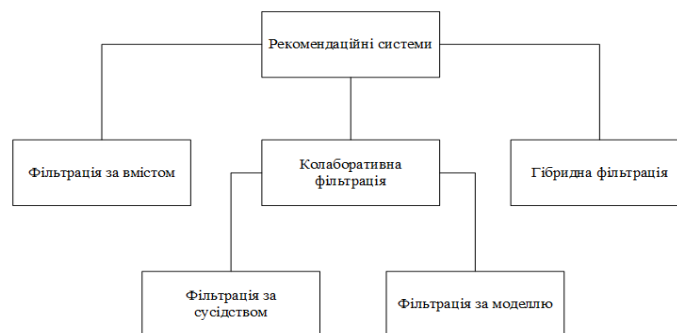


Рис. 1. Види фільтраційних алгоритмів рекомендаційних систем

Колаборативна фільтрація. Алгоритми колаборативної фільтрації працюють за збором відгуків користувача у вигляді рейтингу для елементів і використовують їх при визначенні спільної поведінки оцінювань між кількома користувачами для того, щоб визначити, який елемент порекомендувати. Колаборативну фільтрацію можна далі поділити на підвиди, як зазначено на рисунку 1: *фільтрація за сусідством* і *фільтрація за моделлю*. Розглянемо ці алгоритми.

Колаборативна фільтрація за сусідством. У підходах, що базуються на сусідстві, підмножина користувачів обирається за їхньою спільністю відносно поточного користувача, і зважена комбінація їх оцінок використовується для створення передбачень для заданого користувача. Більшість кроків алгоритму можна записати таким чином:

1. Присвоїти вагу всім користувачам залежно від їхнього рівня спільності із поточним користувачем;
2. Вибрати k користувачів, які мають найбільше спільного із поточним користувачем (часто називаються *сусідами*);
3. Розрахувати передбачення з ваговою комбінації з обраних оцінок сусідів.

У кроці 1 вага $w_{a,u}$ являється мірою спільності між користувачем u та поточним користувачем a . Найбільш поширеною мірою спільності є коефіцієнт кореляції Пірсона. Він записується такою формулою:

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \underline{r}_a)(r_{u,i} - \underline{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \underline{r}_a)^2 \sum_{i \in I} (r_{u,i} - \underline{r}_u)^2}} \quad (1)$$

де I – множина елементів оцінена обома користувачами, $r_{u,i}$ – оцінка, надана елементу i користувачем u і \underline{r}_a – середня оцінка, надана користувачем u [8].

У кроці 3 передбачення розраховуються як зважене середнє значення відхилень від середнього значення сусіда:

$$p_{a,i} = \underline{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \underline{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}} \quad (2)$$

де $p_{a,i}$ – передбачення для поточного користувача a для елемента i , $w_{a,u}$ – подібність між користувачами a та u і K – це сусіди або множина найбільш подібних користувачів [8].

Подібність на основі кореляції Пірсона визначає, до якої міри є лінійна залежність між двома змінними. Інакше оцінки двох користувачів можуть сприйматися як вектор m -розмірного простору і визначити подібність, використовуючи косинус кута, що знаходиться між ними:

$$w_{a,u} = \cos \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2 \times \|\vec{r}_u\|_2} = \frac{\sum_{i=1}^m r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^m r_{a,i}^2} \sqrt{\sum_{i=1}^m r_{u,i}^2}} \quad (3)$$

Потрібно брати до уваги, коли розраховується косинусова подібність, не може бути від'ємних рейтингів, а неоцінені елементи мають рейтинг 0 [8].

Розширення до колаборативної фільтрації за сусідством. У методі колаборативної фільтрації за сусідством є декілька розширень. Одним із цих розширень є *фільтрація за елементами*. Розглянемо її детальніше.

Алгоритм колаборативної фільтрації за сусідством, коли застосований до мільйонів користувачів та елементів, погано масштабується, через розрахункову складність пошуку подібних користувачів. Як альтернативу Грег Лінден запропонував колаборативну фільтрацію елемента до елемента, де зіставляються елементи, оцінені користувачем до подібних елементів, аніж подібні користувачі, як це робиться у звичайній колаборативній фільтрації за сусідством. На практиці використання такого підходу призводить до пришвидшення роботи рекомендаційних систем і до покращення результатів рекомендацій.

У цьому підході подібності між парами елементів i та j розраховуються, використовуючи кореляцію Пірсона:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \underline{r}_i)(r_{u,j} - \underline{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \underline{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \underline{r}_j)^2}} \quad (4)$$

де U – множина елементів користувачів, які оцінили обидва елементи i та j ; $r_{u,i}$ – оцінка, надана елементу i користувачем u і \underline{r}_i – середня оцінка i -го елемента серед користувачів [8].

Тоді рейтинг для i -го елемента для користувача a може бути спрогнозований, використовуючи просте зважене середнє значення:

$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|} \quad (5)$$

де K – множина сусідів k -х елементів, оцінені користувачем a , що найбільш подібні до i -го елемента [8].

Для колаборативної фільтрації за елементами може бути застосована альтернативна метрика, наприклад, скорегована косинусова подібність.

Колаборативна фільтрація за моделлю. Розробка та розвиток моделей (машинне навчання, алгоритми добування даних) дозволяють системам розпізнавати складні шаблони, ґрунтуючись на тренувальних даних, і, після цього, робити розумні передбачення для тестових чи реальних даних. Алгоритми колаборативної фільтрації за моделлю, такі як: Басові моделі, кластерні моделі, були досліджені для усунення недоліків попередніх алгоритмів [12].

Басовий алгоритм колаборативної фільтрації за моделлю. Басова мережа – це орієнтований, ациклічний граф з триплетом $\langle N, A, \theta \rangle$, де кожне ребро $n \in N$ становить випадкову змінну, а кожна направлена дуга $a \in A$ між ребрами – це імовірнісний зв'язок між змінними і θ – це таблиця умовної ймовірності, що розраховує наскільки ребро залежить від його батьків. Басові мережі часто використовуються для завдань класифікації.

Алгоритм колаборативної фільтрації просто Басова використовує стратегію наївного Басова для здійснення передбачень. Припускаючи, що ознаки є незалежними для кожного класу, ймовірність, що певний клас, якому надані всі ознаки, може бути розрахований, і тоді клас з найвищою ймовірністю буде класифікований як спрогнозований клас. Для неповних даних розрахування ймовірності й виконання класифікації визначаються за спостережними даними [12]:

$$class = \arg \arg p(class_j) \prod_0 P(X_0 = x_0 | class_j) \quad (6)$$

Оцінювач Лапласа використовується для розгладження ймовірності розрахунків і уникнення умовної ймовірності 0:

$$P(X_i = x_i | Y = y) = \frac{\#(X_i = x_i, Y = y) + 1}{\#(Y = y) + |X_i|}, \quad (7)$$

де $|X_i|$ – це розмір множини класу $\{X_i\}$ [12].

Мультикласові дані спочатку переводяться у дані бінарного класу і потім перетворюються у булеановий вектор ознак рейтингової матриці. Такі перетворення дозволяють виконувати завдання для колаборативної фільтрації, використовуючи алгоритм наївного Баеса, легше, але через це з'являються проблеми із масштабуванням і втратою мультикласової інформації для мультикласових даних.

Також при спробі застосувати алгоритм простого Баеса для завдань колаборативної фільтрації для мультикласових даних було виявлено, що простий Баес має гіршу точність, але кращу масштабованість за колаборативну фільтрацію, що основана на кореляції Пірсона, оскільки передбачення на кореляції Пірсона базуються на спостережних рейтингах, а час на створення передбачення є менший.

Кластерний алгоритм колаборативної фільтрації за моделлю. Кластер – це колекція об'єктів даних, які подібні один до одного в одному кластері і неподібні до об'єктів у інших кластерах. Міра подібності об'єктів визначається, використовуючи такі метрики: метрика Мінковського та кореляція Пірсона.

Для двох об'єктів даних, $X = (x_1, x_2, \dots, x_n)$ та $Y = (y_1, y_2, \dots, y_n)$, метрика Мінковського визначається так:

$$d(X, Y) = \sqrt[q]{\sum_{i=0}^n |x_i - y_i|^q}, \quad (8)$$

де n – це розмірність об'єкта, x_i та y_i – значення i -го простору об'єкта X та Y відповідно і q – додатне ціле число (коли $q = 1$ – це *відстань Манхеттена*, коли $q = 2$ – це *відстань Евкліда*) [12].

Кластерні методи можуть бути класифіковані в три категорії: *метод розподілу*, *методи на основі щільності* та *ієрархічні методи*. Поширеним методом розподілу є метод k -середніх, запропонований Макквіном, який має дві основні переваги – відносна ефективність і легка імплементація. Методи на основі щільності зазвичай здійснюють пошук щільних кластерів об'єктів, розділених розрідженими регіонами, що представляють шум. DBSCAN та OPTICS – відомі алгоритми з кластерних методів на основі щільності. З відомих ієрархічних алгоритмів є BIRCH. Ці алгоритми створюють ієрархічну декомпозицію множини об'єктів даних за якимось критерієм. Здебільшого кластеризація – це проміжний крок і результуючі кластери використовуються для подальшого аналізу чи обробляються для проведення класифікації чи інших задач. Кластеризація моделей колаборативної фільтрації може бути застосована в різних ситуаціях. Наприклад, кластерна методика використовується для розділення даних у кластери і виконується якийсь метод колаборативної фільтрації, оснований не на моделі, такий як кореляція Пірсона, і створюються передбачення для задач всередині кожного кластера.

Фільтрація за вмістом. Чисті рекомендаційні системи з використанням колаборативної фільтрації використовують тільки матриці оцінок користувачів. Цей підхід відноситься до користувачів та елементів як до атомарних одиниць, де передбачення здійснюються без індивідуальних особливостей користувача чи елемента. Але можна покращити індивідуальну рекомендацію, знаючи більше про користувача, наприклад, демографічну інформацію; або про елемент, наприклад, інформація про продюсера та жанр музичного альбому. Рекомендаційні системи за вмістом надають рекомендації, порівнюючи контент, що описує елемент, з контентом, що цікавить користувача.

Багато досліджень у цій сфері сфокусовано на рекомендації предметів з асоційованим текстовим вмістом, такий як: вебсторінки, альбоми, книги, фільми; де вебсторінки самі чи асоційований контент у вигляді опису чи відгуку користувача доступні. Декілька підходів вирішили цю проблему як завдання отримання інформації (OI), де вміст асоціюється з уподобаннями користувача і розглядається як запит та неоцінені документи розраховуються з актуальністю / подібністю до цього запиту.

Альтернативний підхід до OI – це розглядати рекомендацію як завдання класифікації, де кожен приклад представляє вміст елемента і минулі оцінки користувача використовуються як мітки для цих прикладів.

Гібридна фільтрація. З метою отримати по максимуму переваг з рекомендаційних систем, що використовують фільтрацію за вмістом та колаборативну фільтрацію, було запропоновано декілька гібридних підходів, що об'єднують найкраще з двох алгоритмів. Один простий підхід – це дозволити колаборативній фільтрації і фільтрації за вмістом створювати окремі рейтингові списки рекомендацій і потім об'єднувати їхні результати в кінцевий список.

Був запропонований загальний фреймворк для колаборативної фільтрації, посиленої контентом, де передбачення основані на вмісті застосовані для конвертації розрідженої матриці оцінок користувача в повністю заповнену оцінками матрицю, і потім метод колаборативної фільтрації використовується для здійснення рекомендації. Точніше, використовується класифікатор наївного Баеса, натренований на документах, що описують оцінені елементи кожного користувача і заміняють неоцінені елементи на передбачення з цього класифікатора. Використовується результуюча матриця псевдооцінок для знаходження сусідів подібних до поточного користувача і передбачення виконуються за допомогою

кореляції Пірсона, що правильно зважена для прийняття до уваги перекриття оцінених елементів і для передбачень контенту для поточного користувача. Цей алгоритм працює краще за чисту колаборативну фільтрацію, фільтрацію за вмістом та лінійну комбінацію цих двох.

Існують деякі інші гібридні алгоритми, що основані на колаборативній фільтрації, але також зберігають фільтрацію за вмістом для кожного користувача. Така фільтрація за вмістом використовується для знаходження подібних користувачів. Для прикладу, кожен профіль користувача представлений як вектор зважених слів, що походять з вдалих прикладів тренування, використовуючи алгоритм Уіноу. Передбачення створюються, застосувавши колаборативну фільтрацію прямо до матриці профілів користувачів (на відміну від матриці оцінок користувачів).

Також деякі гібридні підходи намагаються прямо комбінувати вміст та колаборативні дані під імовірнісним фреймворком. Була розширена аспектна модель Хофманна для включення спільної появи даних серед користувачів, предметів і вмісту користувачів.

Розробка алгоритму музичної рекомендаційної системи. Після повного аналізу найпопулярніших підходів до розробки алгоритмів рекомендаційних систем можемо перейти до створення своєї музичної рекомендаційної системи. Розглянувши різні алгоритми створення рекомендацій, можемо дійти до висновку, що найкращою є гібридна фільтрація. Гібридна фільтрація поєднує в собі найкраще з колаборативної фільтрації та фільтрації за вмістом і уникає їхніх недоліків. Давайте спробуємо створити алгоритм для музичної рекомендаційної системи.

Нехай користувачу здебільшого подобаються музичні проекти в $g \subset G$ жанрах, і набір користувачів також надає високі оцінки музичним проектам у $g \subset G$ жанрах, тоді g буде метрикою для рекомендації музичних проектів користувачу.

Вхідні дані: користувачі X ; музичні проекти m ; оцінка r ; жанр музичного проекту m_g ; кількість музичних проектів, що будуть рекомендовані (μ).

Вихідні дані: рекомендовані музичні проекти R .

1. Для всіх користувачів обрати побачені (s) та непобачені музичні проекти (s'), асоціацію з кожним жанром ag_j стосовно s' , де $i - з 1 до n$, а $j - з 1 до m$.

2. Розрахувати $score_j = \frac{ag_j}{m}$.

3. Обрати три найбільші $score_j$.

4. Обрати $m'' \subset s$ для i -го користувача за трьома найбільшими $score_j$.

5. Обрати подібність sim_j стосовно m'' .

6. Обрати користувача з найбільш високим sim_j .

7. Обрати m' , враховуючи найвищі три значення $score_j \subset s$ користувача, отриманого у кроці 6.

8. Розрахувати вагу $W(m_e')$, де $e \subset m'$.

9. Повернути μ найкращих зважених рекомендацій.

Примітка: ag_j – це загальна кількість музичних проектів, що належать до жанру j .

Діаграма класів музичної рекомендаційної системи. Створимо діаграму класів для майбутньої музичної рекомендаційної системи. Вона зображена на рисунку 2.

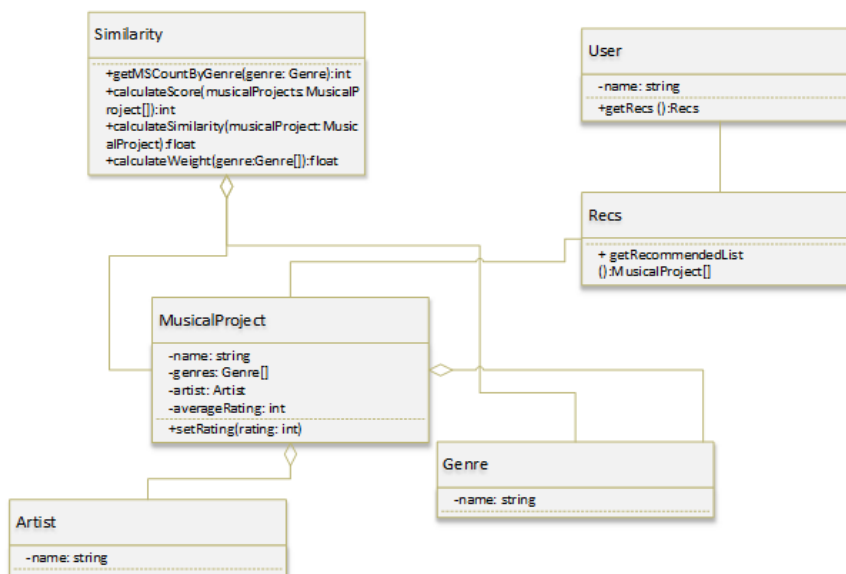


Рис. 2. Діаграма класів музичної рекомендаційної системи

На цій діаграмі можна побачити ряд таких класів:

- *User* – відповідає користувачеві. На ньому є такі властивості та методи:
 - *name* – властивість, що відповідає за ім'я користувача;
 - *getRecs* – метод, що повертає рекомендації для користувача;
- *Recs* – відповідає рекомендаціям. На ньому є такі властивості та методи:
 - *getRecommendedList* – метод, що створює рекомендації персоналізовано для користувача;
- *MusicalProject* – відповідає музичному проєкту. На ньому є такі властивості та методи:
 - *name* – властивість, що відповідає за назву музичного проєкту;
 - *genres* – властивість, що відповідає за жанри музичного проєкту;
 - *artist* – властивість, що відповідає за виконавця музичного проєкту;
 - *averageRating* – властивість, що відповідає середній оцінці музичного проєкту;
 - *setRating* – метод, що додає оцінку до музичного проєкту;
- *Genre* – відповідає за жанр. На ньому є такі властивості та методи:
 - *name* – властивість, що відповідає за назву жанру;
- *Artist* – відповідає за виконавця. На ньому є такі властивості та методи:
 - *name* – властивість, що відповідає за назву виконавця;
- *Similarity* – відповідає за розрахунок подібності. На ньому є такі властивості та методи:
 - *getMSCountByGenre* – метод, що розраховує кількість музичних проєктів за жанром;
 - *calculateScore* – метод, що розраховує рейтинг для кожного музичного проєкту;
 - *calculateSimilarity* – метод, що розраховує коефіцієнт подібності;
 - *calculateWeight* – метод, що розраховує вагу.

Висновки та перспективи подальших досліджень. У результаті дослідження було проаналізовано найбільш вживані алгоритми музичних рекомендаційних систем. Було визначено, що в цілому всі такі алгоритми можна поділити на 3 великі групи: колаборативна фільтрація, фільтрація за вмістом та гібридна фільтрація. Колаборативна фільтрація є однією з найуспішніших технік створення рекомендаційних систем. Колаборативну фільтрацію можна поділити на підмножини: фільтрація за сусідством та фільтрація за моделлю. Фільтрація за сусідством визначає подібність між користувачем та елементом, а потім розраховує зважену суму оцінок або зважене середнє значення для створення передбачення на значенні подібності. Фільтрація за моделлю використовує моделі, які потрібно тренувати, наприклад, Баєсові мережі, кластерні методи для створення передбачень. Кластерні методи колаборативної фільтрації роблять рекомендації на маленьких кластерах, а не на всіх даних, і таким чином мають кращу масштабованість. Також можливо використовувати фільтрацію за вмістом. Цей алгоритм використовує ознаки предметів для того, щоб порекомендувати користувачеві елементи, які подібні до тих, що подобаються користувачеві. Для того, щоб покращити існуючі алгоритми, була створена гібридна фільтрація. Цей вид фільтрації поєднує в собі найкраще з колаборативної фільтрації і фільтрації за вмістом, а також уникає їх основних недоліків.

Враховуючи попередній аналіз алгоритмів, було вирішено провести проектування власної музичної рекомендаційної системи з урахуванням переваг кожного з розглянутих алгоритмів. Спочатку був описаний загальний алгоритм музичної рекомендаційної системи. Для створення алгоритму використовувався гібридний вид фільтрації. Також після написання приблизного алгоритму була створена діаграма класів для майбутньої рекомендаційної системи. В ній знаходяться усі необхідні класи, за допомогою яких буде створюватися музичної рекомендаційної системи.

References:

1. IFPI GLOBAL MUSIC REPORT (2022), [Online], available at: <https://globalmusicreport.ifpi.org/>
2. Roy, D. and Dutta, M. (2022), «A systematic review and research perspective on recommender systems», *Journal of Big Data*, Vol. 9, [Online], available at: <https://cutt.ly/8Moa5FO>
3. Bell, R.M., Bennett, J., Koren, Y. and Volinsky, C. (2009), «The million dollar programming prize», *IEEE Spectrum*, [Online], available at: <https://cutt.ly/CMoaM59>
4. Hosanagar, K. (2007), «Reinforcing the Blockbuster Nature of Media: The Impact of Online Recommenders», *Knowledge at Wharton*, No. 3, [Online], available at: <https://ieeexplore.ieee.org/abstract/document/4907383>
5. Vozalis, E. and Margaritis, K.G. (2003), *Analysis of Recommender Systems' Algorithms*, [Online], available at: https://www.researchgate.net/publication/27383115_Analysis_of_Recommender_Systems'_Algorithms
6. Sahu, S.P., Nautiyal, A. and Prasad, M. (2017), *Machine Learning Algorithms for Recommender System – a comparative analysis*, [Online], available at: <https://cutt.ly/hNHpDMu>
7. Adiyansjah, Gunawan, A.A.S. and Suhartono, D. (2019), «Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks», *Procedia Computer Science*, Vol. 157, pp. 99–109, [Online], available at: <https://cutt.ly/sNHamp9>
8. Melville, P. and Sindhvani, V. (2010), *Recommender Systems*, [Online], available at: <https://www.ime.usp.br/~jstern/miscellanea/seminario/Melville1.pdf>

9. Resnick, P. and Varian, Hal R. (1997), «Recommender systems», *Communications of the ACM*, March, No. 40 (3), pp. 56–58, [Online], available at: <https://cutt.ly/9MosuOS>
10. Burke, R. (2002), «Hybrid Recommender Systems: Survey and Experiments», *User Modeling and User-Adapted Interaction*, November, No. 12 (4), [Online], available at: <https://cutt.ly/pNHshBy>
11. Adomavicius, G. and Tuzhilin, A. (2005), «Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions», *IEEE Transactions on Knowledge and Data Engineering*, New York University, July, No. 17 (6), pp. 734–749, [Online], available at: <https://cutt.ly/ZNHsT53>
12. Xiaoyuan, Su and Khoshgoftaar, T.M. (2009), «Survey of Collaborative Filtering Techniques», *Advances in Artificial Intelligence*, [Online], available at: <https://downloads.hindawi.com/archive/2009/421425.pdf>

Гордєєв Ростислав Сергійович – студент факультету інформаційно-комп’ютерних технологій Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-7179-021X>.

Наукові інтереси:

- вебтехнології;
- штучний інтелект.

Граф Марина Сергіївна – PhD, завідувач кафедри комп’ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-4873-548X>.

Наукові інтереси:

- інтелектуальні системи;
- веборієнтовані технології та аналіз даних.

Hordeiev R.S., Graf M.S.

Analysis of existing algorithms of music recommendation systems

We live in a time characterized by excessive information overload. For example, a user looking for music, goods or videos does not intend to spend a lot of time and delve into the complexities of the search process. In such situations, it is advisable to use recommender systems. Such systems provide a personalized list of items that best meet the user's information needs. Music is one of the most popular areas where the recommender system is used. When visiting any music streaming service (such as Spotify), the user immediately sees a list of recommended songs that they might like. Such systems are quite common on the Web and allow users to save time and effort when searching for the necessary material. Under the hood of such recommender systems, different algorithms can be used, which can be divided into three large groups: collaborative filtering, content filtering, and hybrid filtering. Each algorithm has its own features and use cases, which are considered in a more detailed description of these methods.

After a detailed analysis, we designed our own music recommendation system. An algorithm for creating the system has been developed, which is based on a preliminary analysis of already existing algorithms. To describe the entities of the music recommendation system and their relations, we designed a corresponding class diagram.

Keywords: music recommendation system; collaborative filtering; content filtering; hybrid filtering; collaborative filtering by model; collaborative neighborhood filtering; algorithm; recommendation.

Стаття надійшла до редакції 10.10.2022.