# 7.6. Рекомендаційні системи

Завдання рекомендаційної системи - проінформувати користувача про товар, який йому може бути найбільш цікавий в даний момент часу. Клієнт отримує інформацію, а сервіс заробляє на наданні якісних послуг. Послуги - це не обов'язково прямі продажі пропонованого товару. Сервіс також може заробляти на комісійних або просто збільшувати лояльність користувачів, яка потім виливається в рекламні та інші доходи.

Залежно від моделі бізнесу рекомендації можуть бути його основою, як, наприклад, у [TripAdvisor](https://www.tripadvisor.com/) (https://www.tripadvisor.com/), а можуть бути просто зручним додатковим сервісом (як, наприклад, в інтернет-магазині одягу), що покращує Customer Experience і робить навігацію по каталогу більш зручною.

Персоналізація онлайн-маркетингу - очевидний тренд останнього десятиліття. За оцінками [McKinsey](https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers), 35% виручки Amazon або 75% Netflix припадає саме на рекомендовані товари і відсоток цей, ймовірно, буде зростати. Рекомендаційні системи - це про те, що запропонувати клієнту, щоб зробити його щасливим.

### Основні характеристики рекомендаційних систем

#### 1. Предмет рекомендації - що рекомендується.

Рекомендувати можна що завгодно: товари (Amazon, Ozon), статті (Arxiv.org), новини (Surfingbird, Яндекс.Дзен), зображення (500px), відео (YouTube, Netflix), людей (Linkedin, LonelyPlanet), музику (Last.fm, Pandora), плейлисти та інше.

#### 2. Мета рекомендації - навіщо рекомендується.

Наприклад: покупка, інформування, навчання, налагодження контактів.

#### 3. Контекст рекомендації - що користувач в цей момент робить.

Наприклад: дивиться товари, слухає музику, спілкується з людьми.

#### 4. Джерело рекомендації - хто рекомендує:

* Аудиторія (середній рейтинг ресторану в TripAdvisor).
* Схожі за інтересами користувачі.
* Експертне співтовариство (коли мова про складний товар, наприклад, вино).

#### 5. Ступінь персоналізації.

**Неперсональні рекомендації** - коли користувачеві рекомендують те ж саме, що й всім іншим. Допускається таргетинг за регіоном або часом, але не враховуються особисті переваги користувача.

Більш просунутий варіант - коли рекомендації використовують дані з поточної сесії. Користувач подивився кілька товарів, і внизу сторінки йому пропонуються подібні.

**Персональні рекомендації** використовують всю доступну інформацію про клієнта, в тому числі історію його покупок.

#### 6. Прозорість.

Люди більше довіряють рекомендації, якщо розуміють, як саме вона була отримана. Так менше ризик нарватися на «недобросовісні» системи, які просувають проплачений товар або ставлять більш дорогі товари вище у рейтингу. Крім того, хороша рекомендаційна система сама повинна вміти боротися з проплаченими відгуками і збільшенням рейтингу продавцями.

Маніпуляції бувають і ненавмисними. Наприклад, коли виходить новий бойовик, насамперед на нього йдуть фанати, відповідно, першу пару місяців рейтинг може бути сильно завищений.

#### 7. Формат рекомендації.

Це може бути спливаюче вікно, що з'являється в певному розділі сайту, відсортований список, стрічка внизу екрана або щось інше.

#### 8. Алгоритми.

Незважаючи на велику кількість існуючих алгоритмів, вони зводяться до кількох базових підходів. До найбільш класичних відносяться алгоритми Summary-Based (неперсональний), Content-Based (моделі, що засновані на описі товару), Collaborative Filtering (колаборативна фільтрація), Matrix Factorization (методи засновані на матричному розкладанні) та інші.

У центрі будь-якої рекомендаційної системи знаходиться матриця переваг. Це матриця, в якій по одній з осей відкладено всі клієнти сервісу (Users), а по інший - об'єкти рекомендації (Items). На перетині певних пар (User, Item) дана матриця заповнена оцінками (Ratings) - це показник зацікавленості користувача в даному товарі, виражений за заданою шкалою (наприклад від 1 до 5).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Товар1 | Товар2 | Товар3 | Товар4 | Товар5 |
| Клієнт1 |  | 3 |  | 5 |  |
| Клієнт2 | 1 |  | 1 | 1 |  |
| Клієнт3 | 2 |  |  | 3 | 2 |
| Клієнт4 |  | 4 |  |  | 5 |
| Клієнт5 | 5 |  | 2 |  |  |

Користувачі, зазвичай, оцінюють лише невелику частину товарів, що є в каталозі, і завдання рекомендаційної системи - узагальнити цю інформацію і передбачити ставлення клієнта до інших товарів, про які нічого не відомо. Іншими словами потрібно заповнити всі незаповнені клітинки.

Шаблони споживання у людей різні, і не обов'язково повинні рекомендуватися нові товари. Можна показувати повторні позиції, наприклад, для поповнення запасу. За цим принципом виділяють дві групи товарів.

* **Повторювані**. Наприклад, шампуні або шкарпетки, які потрібні завжди.
* **Неповторювані**. Наприклад, книги або фільми, які не часто придбають повторно.

Якщо продукт не можна явно віднести до одного з класів, має сенс визначати допустимість повторних покупок індивідуально (хтось ходить в магазин тільки заради товару певної марки, а кому-то важливо спробувати все, що є в каталозі).

Поняття «цікавинки» теж суб'єктивне. Деяким користувачам потрібні речі тільки з їхньої улюбленої категорії (Conservative Recommendations), а хтось, навпаки, більше відгукується на нестандартні товари або групи товарів (Risky Recommendations).

Наприклад, відеохостинг може рекомендувати користувачеві тільки нові серії улюбленого серіалу, а може періодично закидати йому нові шоу або взагалі нові жанри. В ідеалі варто вибирати стратегію показу рекомендацій під кожного клієнта окремо, за допомогою моделювання категорії клієнта.

Користувацькі оцінки можна отримати двома способами:

* **Явно (Explicit Ratings)** - користувач сам ставить рейтинг товару, залишає відгук, вподобає сторінку.
* **Неявно (Implicit Ratings)** - користувач явно своє ставлення не висловлює, але можна зробити непрямий висновок з його дій: купив товар - значить він йому подобається, довго читав опис - значить є інтерес тощо.

Звичайно, явні оцінки є кращими - користувач сам говорить про те, що йому сподобалося. Однак, на практиці далеко не всі сайти надають можливість явно виражати свій інтерес, та й не всі користувачі мають бажання це робити. Найчастіше використовуються відразу обидва типи оцінок, що добре доповнюють один одного.

Важливо відрізняти терміни Prediction (прогноз ступеня інтересу) і власне Recommendation (показ рекомендації). Що і як показувати - це окреме завдання, яке використовує отримані на кроці Prediction оцінки, але може бути реалізоване по-різному.

Іноді термін "рекомендація" вживають в ширшому сенсі і розуміють будь-яку оптимізацію, будь то вибірка клієнтів для рекламної розсилки, визначення оптимальної ціни пропозиції або просто вибір найкращої стратегії комунікацій з клієнтом. Класичне визначення даного терміну означає вибір товару, найбільш цікавого для клієнта.

## Неперсоналізовані рекомендації

Неперсоналізовані рекомендації є найпростішими в реалізації. В них потенційний інтерес користувача визначається просто середнім рейтингом товару: «Всім подобається - значить сподобається і вам». За цим принципом працює більшість сервісів, коли користувач не авторизується в системі, наприклад, TripAdvisor.

Показуватися рекомендації можуть по-різному - як банер збоку від опису товару (Amazon), як результат запиту, відсортований за певним параметром (TripAdvisor), або в інший спосіб.

Середній рейтинг від покупців також може зображуватися різними способами. Це можуть бути зірочки поруч з товаром, кількість вподобань, різниця позитивних і негативних голосів (як зазвичай роблять на форумах), частка високих оцінок або взагалі гістограма оцінок. Гістограми - найбільш інформативний спосіб, але в них є один мінус - їх складно порівнювати між собою або сортувати, коли потрібно вивести товари списком.

### Проблема холодного старту

Холодний старт - це типова ситуація, коли ще не накопичено достатню кількість даних для коректної роботи рекомендаційної системи (наприклад, коли товар новий або просто його дуже рідко купують). Якщо середній рейтинг пораховано за оцінками всього трьох користувачів (igor92, xyz\_111 і oleg\_s), така оцінка явно не буде достовірною, і користувачі це розуміють. Часто в таких ситуаціях рейтинги штучно коректують.

1. Перший спосіб - показувати не середнє значення, а згладжене середнє (Damped Mean). Сенс такий: при малій кількості оцінок рейтинг, що відображається більше тяжіє до деякого безпечного «середнього» показника, а як тільки набирається достатня кількість нових оцінок, «усереднене» коригування перестає діяти.
2. Інший підхід - розраховувати по кожному рейтингу інтервали достовірності (confidence Intervals). Математично, чим більше оцінок, тим менше варіація середнього і, отже, більше впевненість в його правильності. А в якості рейтингу можна виводити, наприклад, нижню межу інтервалу (Low CI Bound). При цьому зрозуміло, що така система буде досить консервативною, з тенденцією до заниження оцінок з нових товарів (якщо, звичайно, це не хіт).

Проблема холодного старту також є актуальною і для неперсоналізованих рекомендацій. Загальний підхід тут - замінювати те, що в даний момент не може бути пораховано, різними евристиками (наприклад, замінювати середнім рейтингом, використовувати алгоритм простіше, або взагалі не використовувати товар, поки не будуть зібрано дані).

### Актуальність рекомендацій

У деяких випадках також важливо враховувати «свіжість» рекомендації. Це особливо актуально для статей або постів на форумах. Свіжі записи повинні частіше потрапляти в топ. Для цього використовуються коригувальні коефіцієнти (Damping Factors). Очевидно, що універсальної формули не існує, і кожен сервіс винаходить ту формулу, яка найкраще вирішує його завдання - перевіряється це емпірично.

## Методи створення рекомендаційної системи

Існує два основні підходи до створення рекомендаційних систем - контент-орієнтована та колаборативна фільтрація (рис.1).



Рис.1. Схеми контент-орієнтованої та колаборативної фільтрації

### Content-Based рекомендації

Персональні рекомендації передбачають максимальне використання інформації про самого користувача, в першу чергу про його попередні покупки. Одним з перших з'явився підхід Content-Based Filtering. В рамках даного підходу опис товару (Content) зіставляється з інтересами користувача, отриманими з його попередніх оцінок. Чим більше товар відповідає цим інтересам, тим вище оцінюється потенційна зацікавленість користувача. Очевидна вимога тут - у всіх товарів в каталозі має бути опис.

Історично предметом Content-Based рекомендацій частіше були товари з неструктурованим описом: фільми, книги, статті. Такими ознаками можуть бути, наприклад, текстові описи, рецензії, склад акторів та інше. Однак, можна використовувати і звичайні числові або категоріальні ознаки.

Сontent-Based фільтрація майже повністю повторює механізм Query-Document Matching, який використовується в пошуковій системі Google. Відмінність лише в формі пошукового запиту - тут це вектор, що описує інтереси користувача, а там - ключові слова запитуваної документа. Коли пошукові системи стали додавати персоналізацію, відмінність стерлася ще більше.

### Колаборативна фільтрація (User-Based)

Даний клас систем почав активно розвиватися в 90-і роки. В рамках підходу рекомендації генеруються на підставі інтересів інших схожих користувачів. Такі рекомендації є результатом «колаборації» багатьох користувачів. Звідси і назва методу.

Основне припущення підходу колаборативної фільтрації полягає в тому, що якщо А і В купують аналогічні продукти, А, швидше за все, купить продукт, який купив В ніж продукт, який купила випадкова людина. На відміну від контентно-орієнтованого підходу, тут немає ознак, які відповідають користувачам чи предметам. Алгоритм орієнтується на Матрицю корисності.

1, 2, 3, 4, 5 - користувачі, а стовпці представляють фільми. Значення становлять оцінки (1-5), які користувачі дали фільму. В інших випадках ці значення можуть бути 0/1 залежно від того, чи дивився користувач фільм чи ні.

Рис.2. Матриця корисності в колаборативній фільтрації

Для того щоб алгоритм був ефективний, важливо щоб виконувалося кілька припущень.

* Смаки людей не змінюються з часом (або змінюються, але для всіх однаково).
* Якщо смаки людей збігаються, то вони збігаються у всьому.

Наприклад, якщо два клієнта подобають одні фільми, то книги їм теж подобаються однакові. Так часто буває, коли рекомендовані товари є однорідними (наприклад, тільки фільми). Якщо ж це не так, то у пари клієнтів цілком можуть збігатися переваги в їжі, а політичні погляди бути прямо протилежними - тут алгоритм буде менш ефективним.

У таблиці, де кожен ряд відповідає користувачеві, а кожен стовпець – фільму, просто знайти схожість між рядами в матриці і, відповідно, знайти користувачів із загальними інтересами.

Проте така реалізація пов'язана з низкою проблем:

* Уподобання користувача з часом змінюються. В результаті система може генерувати багато неактуальних рекомендацій.
* Чим більше користувачів, тим більше часу йде на генерування рекомендації.
* Фільтрація користувачів вразлива для штучного збільшення рейтингів, коли зловмисник обманює систему та не об'єктивно підвищує оцінку одних продуктів порівняно з іншими.

Окіл користувача в просторі переваг (його сусіди), який аналізують для генерації нових рекомендацій, можна вибирати по-різному. Можна працювати зі всіма користувачами системи, можна задати певний поріг близькості, можна вибрати кілька сусідів випадковим чином або брати n найбільш схожих сусідів (це найбільш популярний підхід).

Автори з MovieLens як оптимальну кількість сусідів наводять цифри в 30-50 сусідів для фільмів і 25-100 для довільних рекомендацій. Тут зрозуміло, що якщо взяти занадто багато сусідів, то буде більше ймовірність випадкового шуму. І навпаки, якщо взяти занадто мало, то отримуються більш точні рекомендації, але товарів можна рекомендувати меншу кількість.

### Обґрунтування рекомендацій

Важливо, щоб користувач довіряв рекомендаційній системі, а для цього вона повинна бути простою і зрозумілою. За необхідності завжди має бути доступне зрозуміле пояснення рекомендації.

В рамках пояснення добре показувати оцінку товару, що проставили сусіди, з якого саме атрибуту (наприклад, актор чи режисер) був збіг, а також виводити впевненість системи в оцінці (confidence). Щоб не перевантажувати інтерфейс, можна всю цю інформацію винести в кнопку «Докладніше».

Наприклад:

* «Вам може сподобатися фільм ... оскільки там грає ... і ...».
* «Користувачі зі схожими на ваш музичними смаками оцінили альбом ... на 4.5 з 5».

### Оцінка якості системи

Тестування рекомендаційної системи - процес непростий і завжди викликає багато питань, головним чином через неоднозначність самого поняття «якість».

Взагалі, в задачах машинного навчання є два основні підходи до тестування:

* Offline тестування моделі на історичних даних за допомогою ретро-тестів.
* Тестування готової моделі за допомогою A/B тестування (запускається кілька варіантів, вибирається той, що надає кращий результат).

Обидва ці підходи активно застосовуються при розробці рекомендаційних систем.

### Тестування з користувачем

Поведінка користувача погано формалізується і жодна метрика в повній мірі не опише розумові процеси в його голові при виборі товару. На рішення впливає багато факторів. Перехід за посиланням з рекомендованим товаром ще не є його високою оцінкою або навіть інтересом. Частково зрозуміти логіку клієнта допомагає онлайн тестування. Нижче наводиться пара сценаріїв такого тестування.

1. Перший і найочевидніший сценарій - аналіз подій сайту. Що користувач робить на сайті, чи звертає увагу на рекомендації, чи переходить по них, які сервіси системи користуються попитом, які товари краще рекомендуються. Щоб зрозуміти, який з алгоритмів в цілому працює краще або просто спробувати нову перспективну ідею, робиться A/B тестування.
2. Другий сценарій - це отримання відгуків від користувачів у вигляді опитувань і голосувань. Як правило, це загальні питання для розуміння, як клієнти користуються сервісом - що важливіше: релевантність або різноманітність, чи можна показувати повторювані продукти, або це занадто дратує. Перевага сценарію - він надає пряму відповідь на всі ці питання.

Подібне тестування є складним, але для великих рекомендаційних сервісів воно необхідне. Питання можуть бути і більш складними, наприклад, «який зі списків вам здається більш релевантним», «наскільки лист виглядає цілісним», «чи будете ви дивитися цей фільм / читати книгу».

### Неявні рейтинги і унарні дані

На початку свого розвитку рекомендаційні системи застосовувалися в сервісах, де користувач явно оцінює товар шляхом виставлення йому рейтингу - це і Amazon, і Netflix і інші сайти інтернет торгівлі. Однак зі зростанням популярності рекомендаційних систем виникла потреба застосовувати їх ще й там, де жодних рейтингів немає, сервіси, де з якоїсь причини неможливо налагодити систему оцінювання. У цих випадках інтереси користувача можна обчислити лише за непрямими ознаками - про користувацькі переваги говорять певні дії з товаром, наприклад, перегляд опису на сайті, додавання товару в кошик тощо. Тут використовується принцип «купив - значить любить!». Така система неявного оцінювання називається Implicit Ratings.

Неявні рейтинги очевидно працюють гірше явних, оскільки вносять більше шуму. Адже користувач міг купити товар в подарунок дружині або зайти на сторінку з описом товару, тільки щоб залишити там коментар в стилі «яка ж все-таки це гидота» або задовольнити свою природну цікавість.

Якщо у випадку з явними рейтингами можна очікувати, що хоч одну негативну оцінку поставлять, то тут негативної оцінки нема з чого братися. Якщо користувач не купив книгу «П'ятдесят відтінків сірого», він міг це зробити з двох причин:

* Вона йому дійсно не цікава (це негативний кейс).
* Вона йому цікава, але він просто не знає про неї (це пропущений позитивний кейс).

Немає жодних даних, щоб відрізнити перший випадок від другого. Це погано, оскільки навчаючи модель, її слід підкріплювати на позитивних кейсах і штрафувати на негативних, а так в підсумку модель буде зміщеною.

Другий кейс - це можливість залишати тільки позитивні оцінки. Яскравий приклад - це кнопка Like в соцмережах. Рейтинг тут проставляється вже явно, але як і в попередньому прикладі, немає негативних прикладів - відомо, які канали користувачеві подобаються, але не відомо, які не подобається.

## Короткий опис рішення Netflix

[Netflix Prize](https://www.netflixprize.com/) - це конкурс, що проводився в 2009 році, в якому вимагалося спрогнозувати оцінку користувачами фільмотеки Netflix. Непогані призові в 1 млн доларів викликали ажіотаж і залучили велику кількість учасників, в тому числі досить відомих людей з галузі штучного інтелекту.

Це було завдання з явними рейтингами, оцінки ставилися за шкалою від 1 до 5, а точність прогнозу оцінювалася за середньоквадратичною похибкою. Більшість перших місць зайняли великі ансамблі класифікаторів.

[Ансамбль-переможець](https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf) використав моделі наступних класів:

* Basic Model - регресійна модель, заснована на середніх оцінках
* Collaborative Filtering - колаборативна фільтрація
* RBM - обмежені машини Больцмана
* Random Forests – ансамбль дерев рішень.

Як мета-алгоритм, який об'єднував оцінки локальних алгоритмів, використовувався традиційний градієнтний бустінг (метод побудови ансамблю моделей, у якому базові моделі навчаються послідовно і кожна наступна модель ансамблю застосовується до результатів на виході попередньої.).

## Рекомендаційна система Facebook

Рекомендаційна система Facebook визначає, які повідомлення з'являються в стрічці новин конкретного користувача і в якому порядку, передбачаючи, що його найймовірніше зацікавить. Ці прогнози ґрунтуються на тому, за ким стежив користувач, що вподобав і з ким нещодавно спілкувався. Facebook пропонує розібрати принцип роботи алгоритмів рекомендаційної системи на прикладі абстрактного користувача соцмережі ААА.

В основі системи ранжування контенту у Facebook лежить машинне навчання, на базі якого працює стрічка новин. Система ранжирування – це не просто єдиний алгоритм. Це кілька рівнів моделей машинного навчання та рейтингів, які застосовуються для прогнозування контенту, найбільш актуального та значущого для кожного користувача. У міру проходження кожного етапу система ранжування звужує ці тисячі постів-кандидатів до кількох сотень, які з'являються у стрічці новин ААА.

З того часу, як ААА останній раз був онлайн, його друг БББ опублікував фотографію свого собаки, а його подруга ВВВ опублікувала відео з ранкової пробіжки. Сторінка, яка подобається ААА, опублікувала цікаву статтю про те, як найкраще побачити Чумацький Шлях уночі, а його улюблена кулінарна група опублікувала чотири нові рецепти.

Щоб вирішити, які з цих постів мають відображатися вище у стрічці новин ААА, потрібно передбачити, що для нього найважливіше і який контент має для нього найбільшу цінність. З математичної точки зору системі потрібно визначити цільову функцію ААА і виконати однокритеріальну оптимізацію (рис.3).



Рис.3. Рекомендаційна система Facebook

Для цього можуть використовуватися характеристики повідомлення, наприклад, хто відзначений на фотографії і коли вона була опублікована, щоб передбачити, чи вона сподобається ААА. Наприклад, якщо він часто переписується з ВВВ в месенджері або коментарях під постами, а її нове відео було викладено невдовзі після їхнього спілкування, висока ймовірність, що ААА воно сподобається.

Якщо нещодавно ААА дивився більше відеоконтенту, ніж фотографій, його з меншою ймовірністю зацікавить фото собаки БББ. У цьому випадку алгоритм ранжування Facebook видасть відео ВВВ вище, ніж фото собаки БББ.

Щодня люди діляться статтями, які їм цікаві, дивляться відео людей чи знаменитостей, на яких вони підписані, або залишають змістовні коментарі до повідомлень своїх друзів. З математичної точки зору, все стає складніше, коли потрібно оптимізувати видачу під кілька цілей, головна з яких — створити цінність для користувача на довгостроковій основі.

Щоб зрозуміти, що створює довгострокову цінність для людини, Facebook проводить серед користувачів опитування. Потім бере до уваги кожен прогноз ААА, спираючись на відповіді інших користувачів.

Система збирає всі пости-кандидати, які може ранжирувати для ААА (фото собаки, відео, пост про Чумацький шлях і рецепти). Цей арсенал містить весь нещодавно розміщений контент друзів, груп або сторінок, на які підписаний ААА.

Далі системі необхідно оцінити кожну одиницю контенту за її типом і подібністю до контенту, з яким зазвичай взаємодіє ААА. Щоб дати оцінку тисяч постів кожного з 2 млрд користувачів в реальному часі ці моделі запускаються для всіх постів паралельно на кількох нейромережах.

Первинні прогнози потрібні, щоб можна було звузити пул контенту для ранжування. На наступному етапі спрощена модель звужує коло кандидатів приблизно до 500 постів для ААА. Ранжування меншої кількості постів дозволяє використовувати потужніші моделі нейронних мереж для наступних проходів.

Далі йде основний етап підрахунку очок, у якому відбувається переважна більшість персоналізації. Тут оцінка кожної історії розраховується незалежно, всі 500 постів упорядковуються за кількістю оцінок. Для деяких оцінка може бути вищою за вподобання, ніж за коментарі, оскільки деяким людям більше подобається висловлювати свою перевагу вподобаннями, ніж за коментарями.

Зрештою, запускається контекстний фільтр, де спрацьовують функції аналізу різноманітності типів контенту. Вони повинні перевірити, чи в стрічці новин ААА з'явиться досить різноманітний контент, і система не видасть кілька відеопостів один за одним. Всі ці етапи ранжирування відбуваються за час, необхідний ААА, щоб відкрити програму Facebook. За кілька секунд він бачить стрічку новин, яка містить оптимальний для нього набір контенту.

## Висновки

Постановка завдання генерації рекомендацій дуже проста - складається матриця переваг з відомими оцінками користувачів, ці оцінки доповнюються інформацією по клієнту і товару, система спромагається заповнити невідомі значення.

Незважаючи на простоту постановки, виходять сотні статей, що описують принципово нові методики її вирішення. По-перше це пов'язано з ростом кількості зібраних даних, які можна використовувати в моделі і збільшенням ролі implicit-рейтингів. По-друге, з розвитком глибокого навчання і появою нових архітектур нейронних мереж. Все це кратно збільшує і складність моделей.

## Контрольні запитання

1. Назвати основні причини, за якими сервіси застосовують системи рекомендацій.
2. Назвати суттєві характеристики рекомендаційних систем.
3. В чому полягає різниця явного і неявного збору даних про користувача?
4. Яким чином можна отримати користувацьку оцінку?
5. Навести приклади популярних ресурсів, де спостерігається вплив рекомендаційної системи на запити користувача.
6. Як можна вирішити проблему «холодного старту» в рекомендаційних системах?
7. Як можна оцінити роботу рекомендаційної системи?
8. Як рекомендаційна система Facebook формує стрічку новин?

## Літературні джерела

1. Анатомія рекомендаційних систем. Частина перша <https://habr.com/company/lanit/blog/420499/>
2. Анатомія рекомендаційних систем. Частина друга <https://habr.com/ru/company/lanit/blog/421401/>
3. Рекомендательные системы: идеи, подходы, задачи <https://habr.com/ru/company/jetinfosystems/blog/453792/>
4. Сучасні рекомендаційні системи <https://proglib.io/p/sovremennye-rekomendatelnye-sistemy-2021-03-02>
5. Глибоке занурення в рекомендаційну систему Netflix
<https://ichi.pro/ru/glubokoe-pogruzenie-v-rekomendatel-nuu-sistemu-netflix-57277795941192>
6. Штучний інтелект при формуванні персональних рекомендацій <https://habr.com/ru/post/560134/>
7. Методи для рекомендаційних систем на основі вмісту <https://ichi.pro/ru/metody-dla-rekomendatel-nyh-sistem-na-osnove-soderzania-111016630465952>
8. Принцип роботи рекомендаційної системи Facebook <https://performance360.ru/facebook-recommendation-system-guide/>
9. Філософія алгоритму: як рекомендаційні системи змінили людство <https://vc.ru/future/204385-filosofiya-algoritma-kak-rekomendatelnye-sistemy-izmenili-chelovechestvo>
10. Як потрапити в рекомендації Інстаграму <https://smmplanner.com/blog/kak-popast-v-riekomiendovannoie-instaghrama/>