

ДОСЛІДЖЕННЯ МЕТОДІВ РОЗРОБКИ ОНЛАЙН-МАГАЗИНІВ ТА РЕКОМЕНДАЦІЙНИХ СИСТЕМ

Керівник: Курочкін А.В.

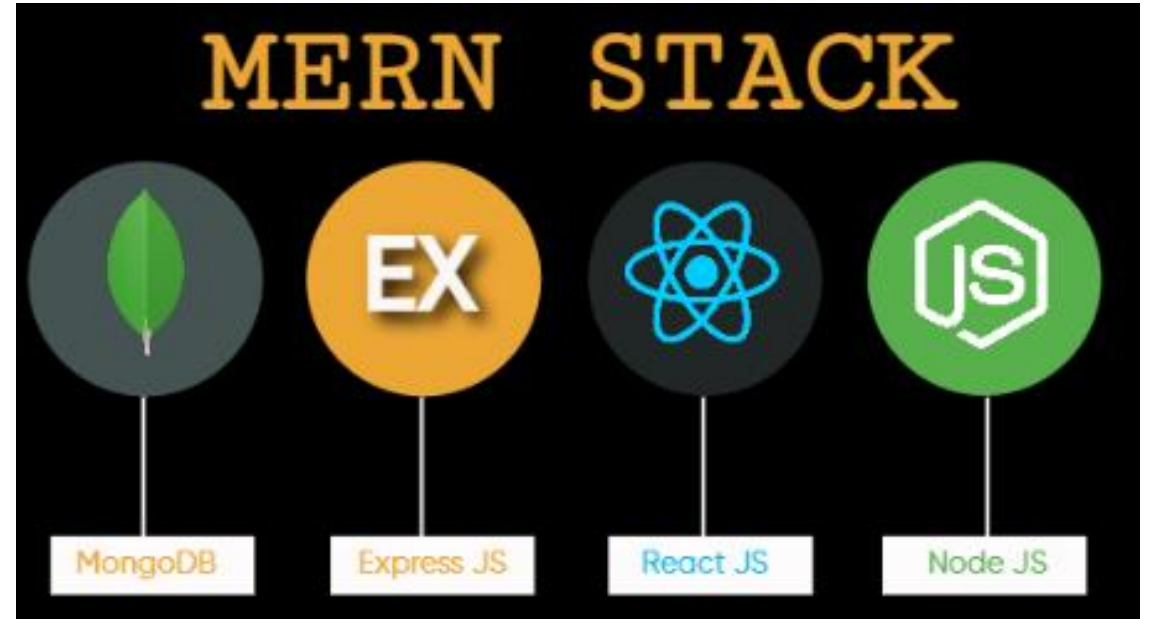
Виконав: Кириченко Євгеній

Завдання

- Дослідити архітектури розробки веб-додатків
- Дослідити різновиди рекомендаційних систем, визначити їх переваги та недоліки
- Створити онлайн-магазин для продажу товарів, пов'язаних з аніме, та рекомендаційної системи для нього

Онлайн-магазин

- React js для клієнтської частини
- Node js та Express js для серверної
- MongoDB база даних

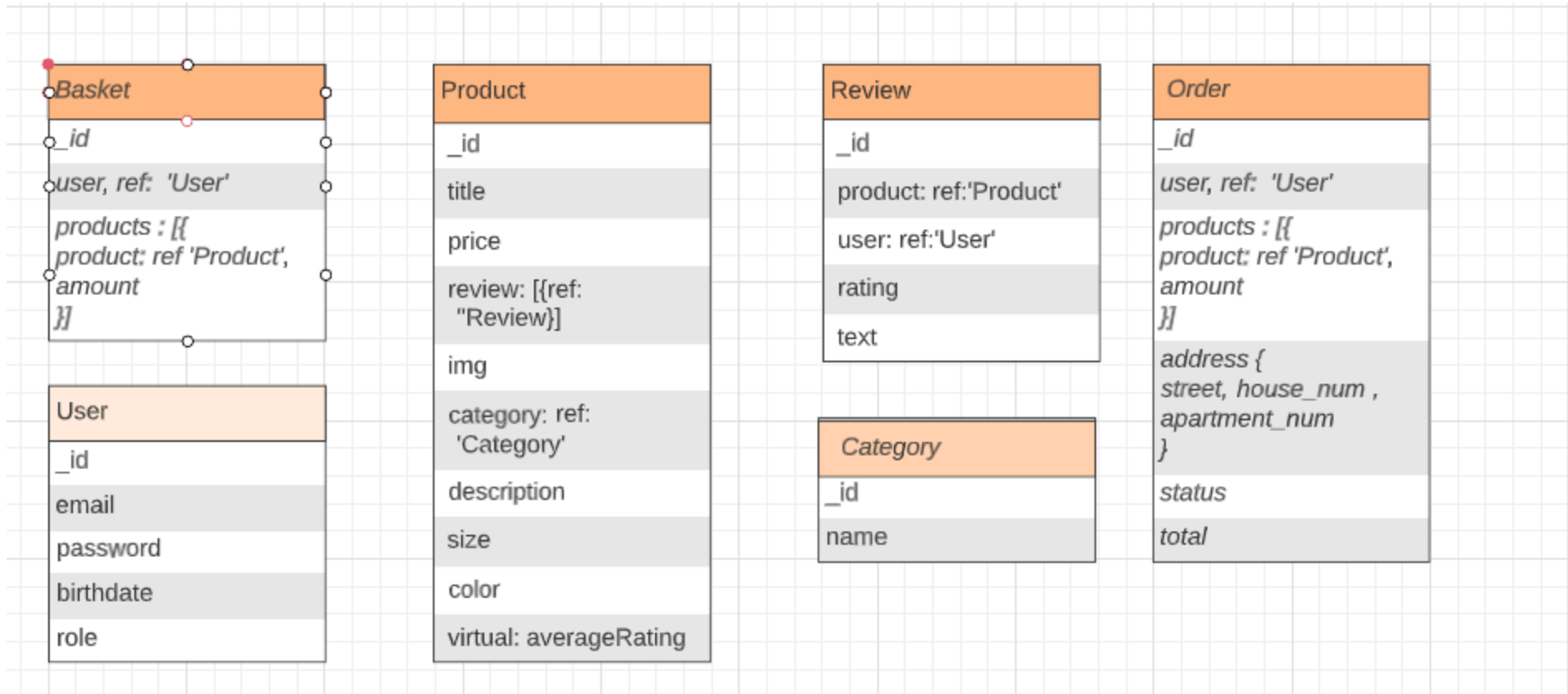


КЛІЄНТ

- Компоненти
- Сторінки
- Маршрутизація
- Сховища
- Запити на сервер

```
export const fetchProducts = async (cat, search, page, limit, priceRange, alphabetOrder) => {  
  const {data} = await $host.get({uri: 'product/', config: {  
    params: {  
      category: cat,  
      search: search,  
      page: page,  
      limit: limit,  
      priceRange: priceRange,  
      alphabetOrder: alphabetOrder  
    }  
  })  
  
  return data  
}
```

Модель даних




Функціонал


- сторінка всіх продуктів з фільтрами
- сторінка продукту
- сторінка схожих продуктів
- сторінка рекомендацій
- кошик, керування кошиком
- оформлення замовлення
- коментарі й оцінки
- функції адміністратора


Інтерфейс(головна сторінка)


The screenshot shows a web interface for selling pants. At the top, there is a navigation bar with a storefront icon on the left and icons for a heart, shopping cart, user profile, 'адмін' (admin), and a back arrow on the right. Below the navigation bar, there are several filters: a green 'reset' button, a 'Category' dropdown menu, a price range '0 - 2000' with a dropdown arrow, a blue-bordered 'a-z' dropdown menu, and a search input field with the placeholder text 'Search'. The main content area displays four product cards, each featuring a pair of pants, a title, a description, a price, a five-star rating, and a 'to cart' button. The products are: 'Naruto pants' (500\$), '1 piece pants' (770\$), 'ball z pants' (720\$), and 'Hunter pants' (550\$). A 'more >' link is visible at the bottom left.

reset Category ▾ 0 - 2000 ▾ a-z ▾ Search


Naruto pants
Штани мають принт з хмарами з аніме "Нар ...
500\$
★★★★★
to cart


1 piece pants
Штани мають принт з образами улюблених п ...
770\$
★★★★★
to cart


ball z pants
Штани мають принт з образами улюблених п ...
720\$
★★★★★
to cart


Hunter pants
Штани мають принт з образами улюблених п ...
550\$
★★★★★
to cart

more >

КОШИК



square_pil

Квадратна подушка з аніме персонаже ...

5



detail

delete



throw_pil

легка та компактна, ідеальна для гр ...

5




detail

delete

Переглянути замовлення

Додавання в кошик



4

Закрити

Додати

to cart

to cart

The image shows a product detail view for a pair of green patterned pants. The pants are displayed on a person's legs. Below the image is a quantity selector with a minus button, the number '4', and a plus button. At the bottom of the product card are two buttons: 'Закрити' (Close) and 'Додати' (Add). The background is a dark grey, and there are 'to cart' buttons visible at the bottom of the page.

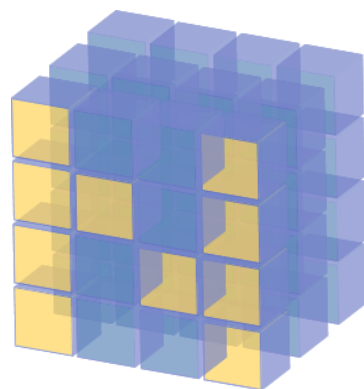
Сторінка адміна для замовлень

ALL ORDERS

	User	Address	Total	Status		
1	jek	Shevchenka 40	1000	pending	Приняти	Видалити
2	jek	Shevchenka 40	1000	pending	Приняти	Видалити
3	jek	Shevchenka 40	1000	pending	Приняти	Видалити
4	jek	Shevchenka 40	1000	pending	Приняти	Видалити
5	jek	Shevchenka 40	1000	pending	Приняти	Видалити
6	jek	Shevchenka 40	1000	pending	Приняти	Видалити
7	jek	Shevchenka 40	1000	pending	Приняти	Видалити
8	jek	Shevchenka 40	1000	pending	Приняти	Видалити
9	jek	Shevchenka 40	1000	pending	Приняти	Видалити
10	jek	Shevchenka 40	1000	pending	Приняти	Видалити
11	jek	Shevchenka 40	1000	pending	Приняти	Видалити
12	jek	Shevchenka 40	1000	pending	Приняти	Видалити
13	jek	Shevchenka 40	1000	pending	Приняти	Видалити

Рекомендаційна система: технології

Python бібліотеки : sklearn, numpy, pandas

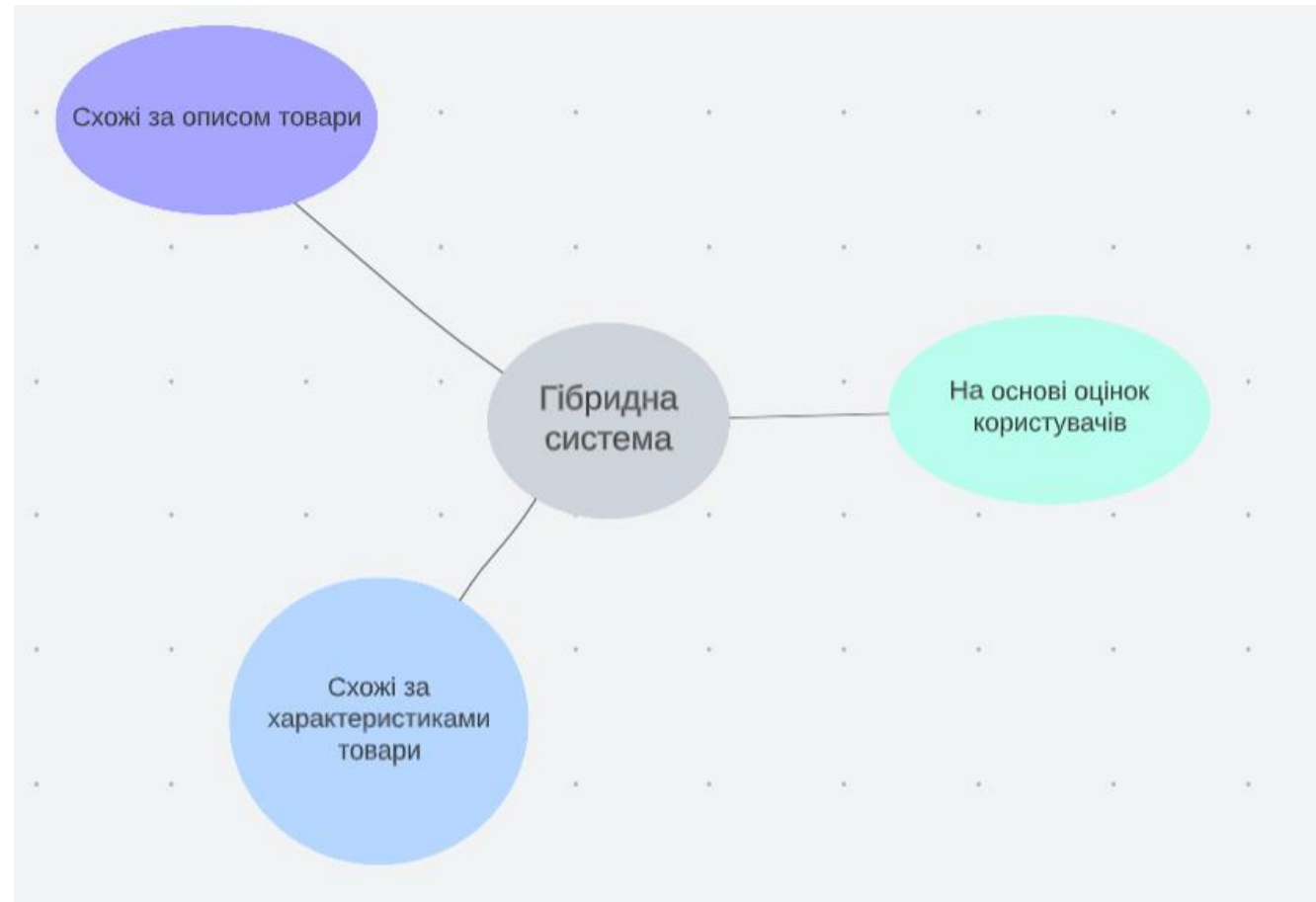


NumPy

Pandas

























Розробка рекомендаційної системи



Collaborative система для рекомендацій (На основі оцінок користувачів)

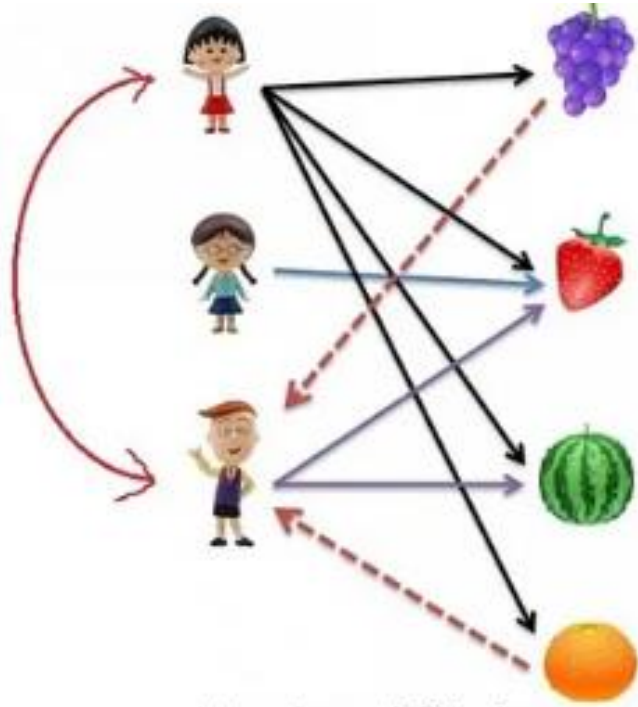
Колаборативні системи створюють рекомендації на основі подібності вподобань користувачів

	 Book 1	 Book 2	 Book 3	 Book 4	 Book 5
 User A					
 User B					
 User C					
 User D					

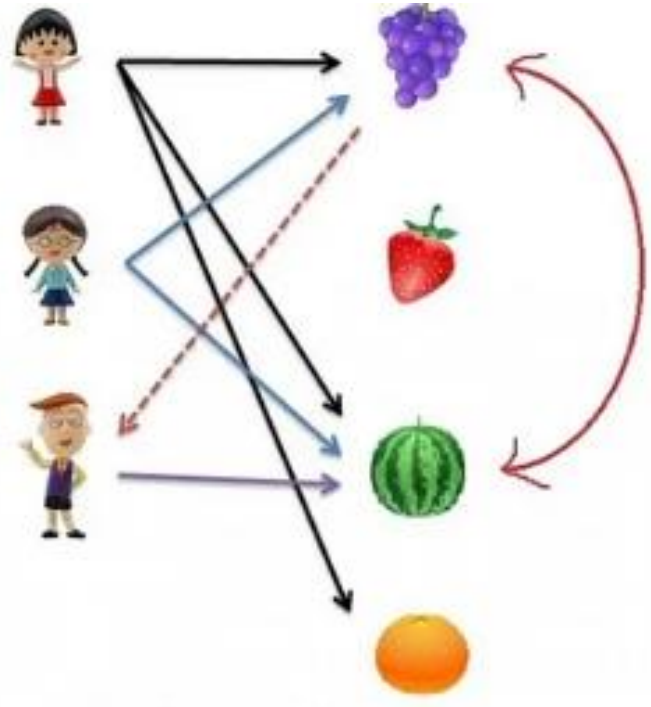
Memory based та Model Based

- Memory based ефективно працюють з невеликими наборами даних, швидко та легко імплементуються.
- працюють на основі метрик подібності
- Model based потребують більше обчислювальних ресурсів
- використовують моделі машинного навчання
- залежить від підбору навчальних даних

Memory-based Collaborative filtering



User-based filtering



Item-based filtering

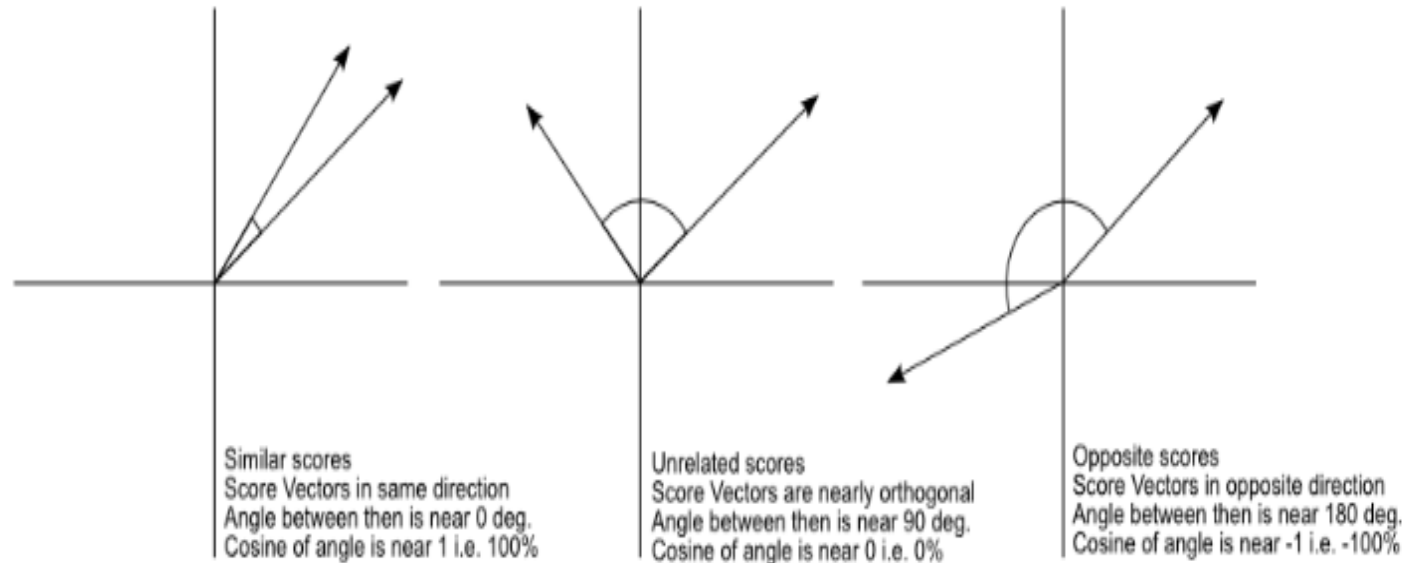
Подібність між користувачами

Розраховується на основі матриці оцінок, за допомогою знаходження подібності між векторами оцінок

	Item1	Item2	Item3	Item4	Item5
User1	3	5	5	1	1
User2	4	5	5	2	3
User3	5	4	2	4	5
User4	3	5	4	3	5
User5	1	4	5	2	5
User6	2	5	4	1	2

Косинусна подібність

Вимірюється косинус кута між векторами. Чим менший кут між векторами, тим більша подібність між користувачами/об'єктами.



Матриця подібності

В результаті ми отримуємо матрицю подібності користувачів, i -й рядок якої - вектор подібності i -того користувача з усіма іншими

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
u_1	0	0.1	-0.2	0.4 (①)	0	0	0	0.1	0.3 (②)	-0.2	0.1	-0.1
u_2	0.1	0	0	-0.1	0	0	0	0.2	0.6	0	0	0
u_3	-0.2	0	0	-0.2	0	0	-0.1	0.4	-0.2	-0.1	0	-0.1
u_4	0.4	-0.1	-0.2	0	0.2	0	0.2	0	0	-0.1	0.6	-0.4
u_5	0	0	0	0.2	0	0	0.3	0	0.2	0	0.3	0
u_6	0	0	0	0	0	0	0	0	0	0.3	0	0
u_7	0	0	-0.1	0.2	0.3	0	0	-0.2	0.5	-0.1	0.2	0.1
u_8	0.1	0.2	0.4	0.0	0	0	-0.2	0	-0.2	-0.2	0.1	-0.2
u_9	0.3	0.6	-0.2	0	0.2	0	0.5	-0.2	0	0	0	-0.2
u_{10}	-0.2	0	-0.1	-0.1	0	0.3	-0.1	-0.2	0	0	-0.1	0.4
u_{11}	0.1	0	0	0.6	0.3	0	0.2	0.1	0	-0.1	0	0
u_{12}	-0.1	0	-0.1	-0.4	0	0	0.1	-0.2	-0.2	0.4	0	0

Weighted mean

Алгоритм для передбачення, як користувач і оцінить об'єкт j

1. З матриці подібності беремо вектор подібності між i -тим користувачем та усіма іншими
2. З матриці оцінок беремо вектор оцінок користувачів об'єкту j
3. Користувачі, які не оцінили об'єкт не враховуються
4. Знаходимо добуток векторів подібності та оцінок, і ділимо його на суму елементів вектора подібності

Weighted Mean Formula

$$\text{Weighted mean} = \frac{\sum_{i=1}^n \text{weight}_i \times x_i}{\sum_{i=1}^n \text{weight}_i}$$

Ілюстрація знаходження weighted mean

↕	0 ↕	1 ↕	2 ↕	3 ↕	4 ↕	5 ↕
0	1.000000	0.701724	0.875308	0.795596	0.701724	0.831252
1	0.701724	1.000000	0.732759	0.675707	1.000000	0.623946
2	0.875308	0.732759	1.000000	0.933119	0.732759	0.772888
3	0.795596	0.675707	0.933119	1.000000	0.675707	0.738367
4	0.701724	1.000000	0.732759	0.675707	1.000000	0.623946
5	0.831252	0.623946	0.772888	0.738367	0.623946	1.000000

Матриця подібності між користувачами.
Беремо вектор подібності першого користувача з усіма іншими. Назвемо його v

↕	0 ↕	1 ↕	2 ↕	3 ↕	4 ↕
0	5	5.0	5.0	5	5.0
1	4	NaN	2.0	4	2.0
2	4	1.0	2.0	4	2.0
3	4	1.0	NaN	4	2.0
4	4	NaN	2.0	4	2.0
5	5	5.0	NaN	5	NaN

Матриця оцінок. Беремо вектор оцінок для першого об'єкта. Назвемо його u

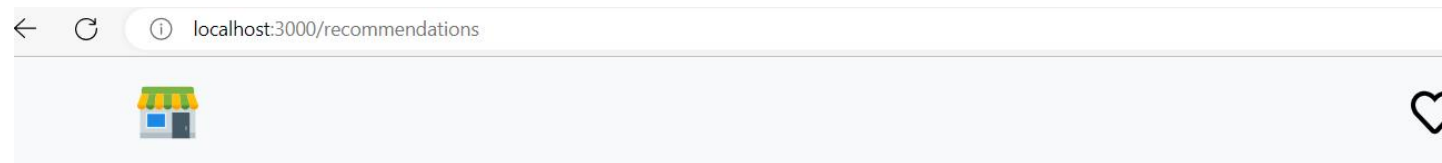
Для знаходження того, як перший користувач оцінить перший об'єкт потрібно добуток векторів u та v поділити на суму елементів вектора v . Передбачена оцінка = $u*v/\text{sum}(v)$

Функція рекомендації

Для кожного продукту знаходимо середнє зважене, після чого рекомендуємо користувачу k товарів з найвищими оцінками

```
def get_top_k_rec(_id, k):  
    rates = []  
    for i in range(0, len(r_matrix.iloc[0])):  
        rates.append(cf_user_wmean(_id, i))  
    sorted_indices = np.argsort(rates)[::-1]  
    sorted_indices = sorted_indices[:k]  
    sorted_pairs = [(i, rates[i]) for i in sorted_indices]  
    return sorted_pairs
```

Використання системи



ghoul hood

це легендарний токійський
Гуль. Дізнайся ...

650\$



to cart



Naruto pants

Штани мають принт з
хмарами з аніме "Нар ...

500\$



to cart



attack titans

це стильний і зручний одяг з
капюшоном, ...

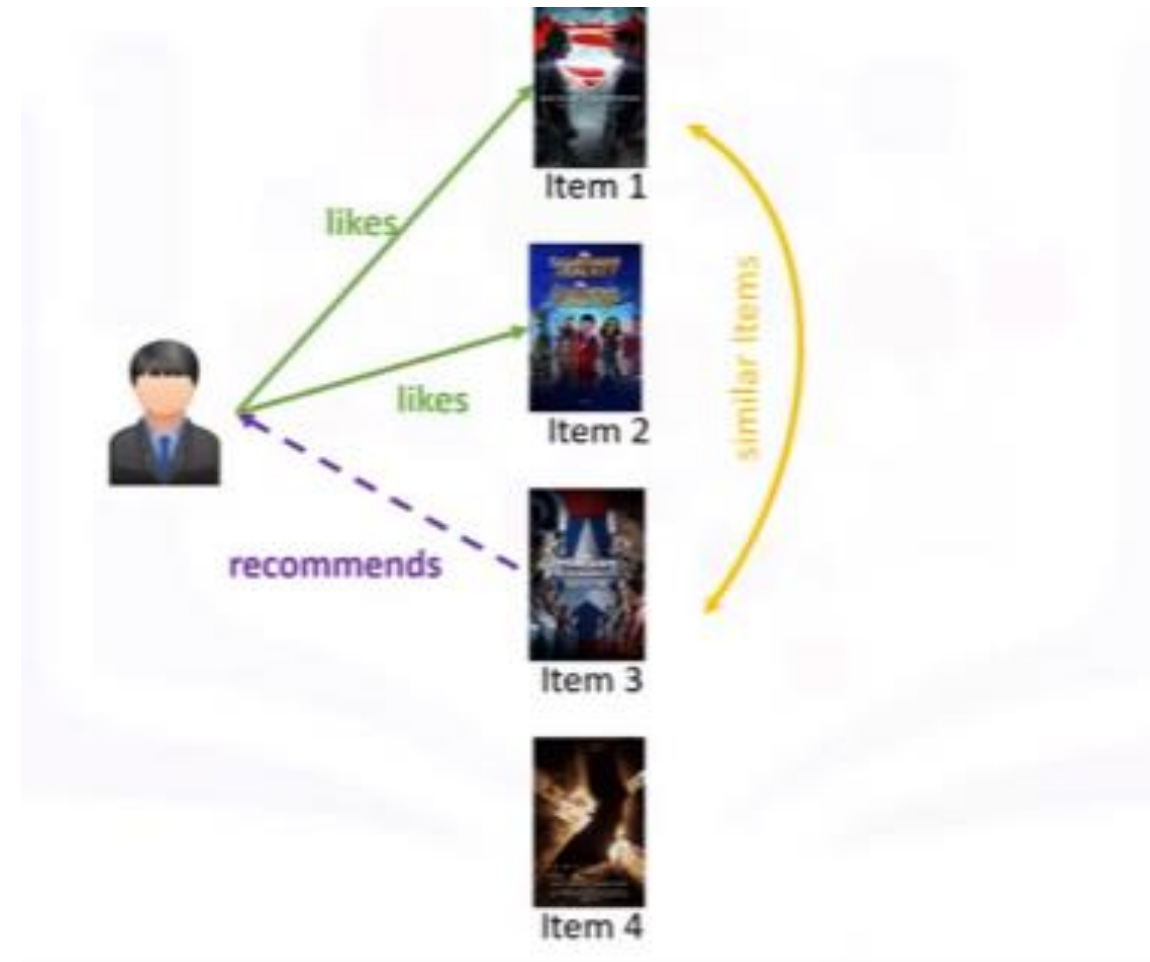
650\$



to cart

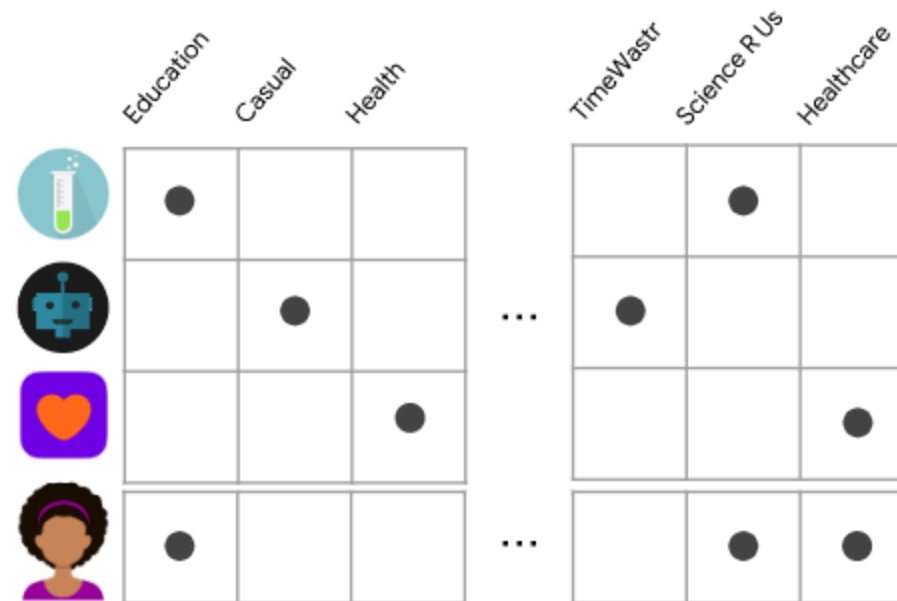
Content-based для товарів (схожі за характеристиками товари)

Задача: обрати к товарів,
найбільш подібних до даного за
характеристиками: опис,
категорія, ціна



Content-based системи

Content-based системи працюють на основі подібності між об'єктами, не враховуючи матриці взаємодій користувачів та об'єктів. Може враховуватись вектор вподобань користувача, який буде співставлятися з векторами характеристик об'єктів.



Як знайти схожі товари

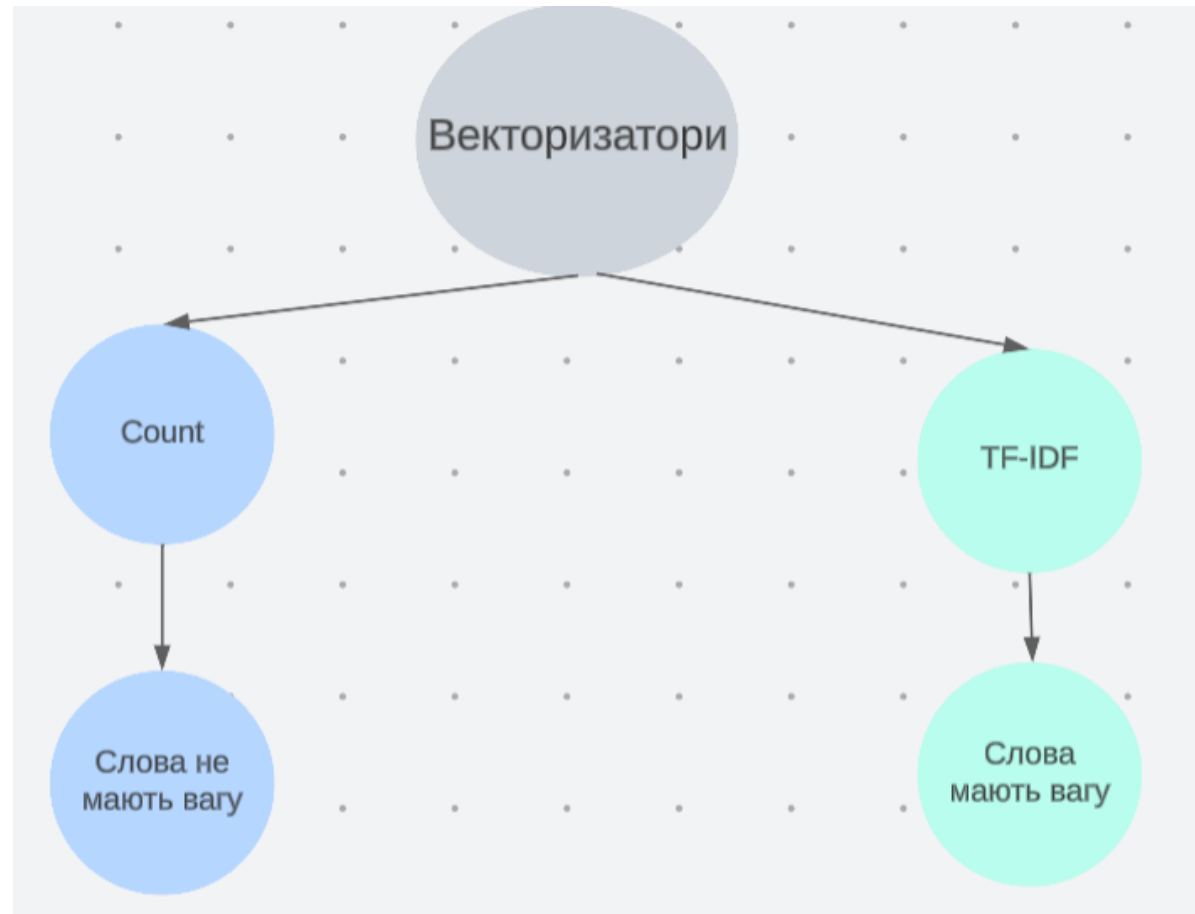
- Набори характеристик товарів переводяться в текстову стрічку
- Текстові стрічки переводяться у вектори
- Після цього знаходимо матрицю подібності між векторами

Векторизатори

Дають змогу сформувати з текстових стрічок вектори, які будуть вміщувати в собі інформацію про те, як часто зустрічається кожне слово.

	big	count	create	dataset	differnt	features	hello	is	james	my	name	notebook	of	python	this	to	try	trying	vectorizer	words
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0	0	0	0	0
2	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
3	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1
4	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0

Найпопулярніші векторизатори



Врахування ціни

Ціну не має сенсу додавати в стрічку для векторизації.

Тому створимо матрицю подібності цін. Значення в ній будуть від 0 до 1, де 0 відсутність подібності, а 1 - однакова ціна

```
def get_price_coeff_matrix(price_list):  
    mean = np.mean(price_list)  
    n = len(price_list)  
    diff_matrix = [[0] * n for _ in range(n)]  
  
    for i in range(n):  
        for j in range(n):  
            m = np.max([1 - abs(price_list[i] - price_list[j]) / (mean * 2), 0])  
            diff_matrix[i][j] = m  
    print(diff_matrix)  
    return diff_matrix
```

Матриця подібності

- Використовуємо `countVectorizer`
- Формуємо вектор (враховуємо категорію та опис)
- Знаходимо матрицю косинусної подібності між векторами u
- Знаходимо матрицю подібності цін v
- Множимо u на v та отримуємо фінальну матрицю подібності

Функція рекомендації

З матриці подібності дістаємо k товарів, найбільш подібних до даного

```
sim_scores = list(enumerate(products_sim[idx]))  
  
sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)  
sim_scores = sim_scores[1:1 + k]
```

Робота системи



адмін



Тоторо футболка

Має принт з персонажем аніме Хаяо Міядза ...

444\$



to cart



Замок футболка

Має принт замок з аніме Хаяо Міядзакі "З ...

565\$



to cart



ball z pants

Штани мають принт з образами улюблених п ...

720\$

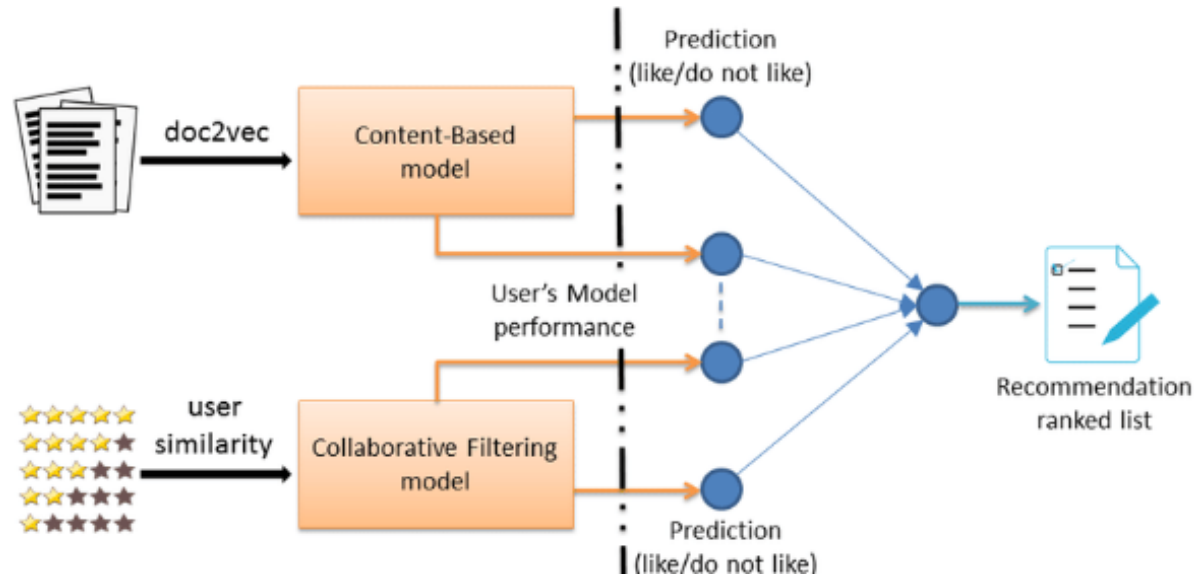


to cart

Гібридна система (схожі за описом товари)

Гібридні системи поєднують в собі collaborative та content-based моделі для досягнення кращого результату.

Завдання: знайти товари, схожі за описом до заданого, які користувач оцінив би високо.



Схожі за описом товари (content-based)

TF-IDF векторизатор - кожне слово має свою вагу

$$w_{i,j} = tf_{i,j} \times \log \left(\frac{N}{df_i} \right)$$

$tf_{i,j}$ = number of occurrences of i in j

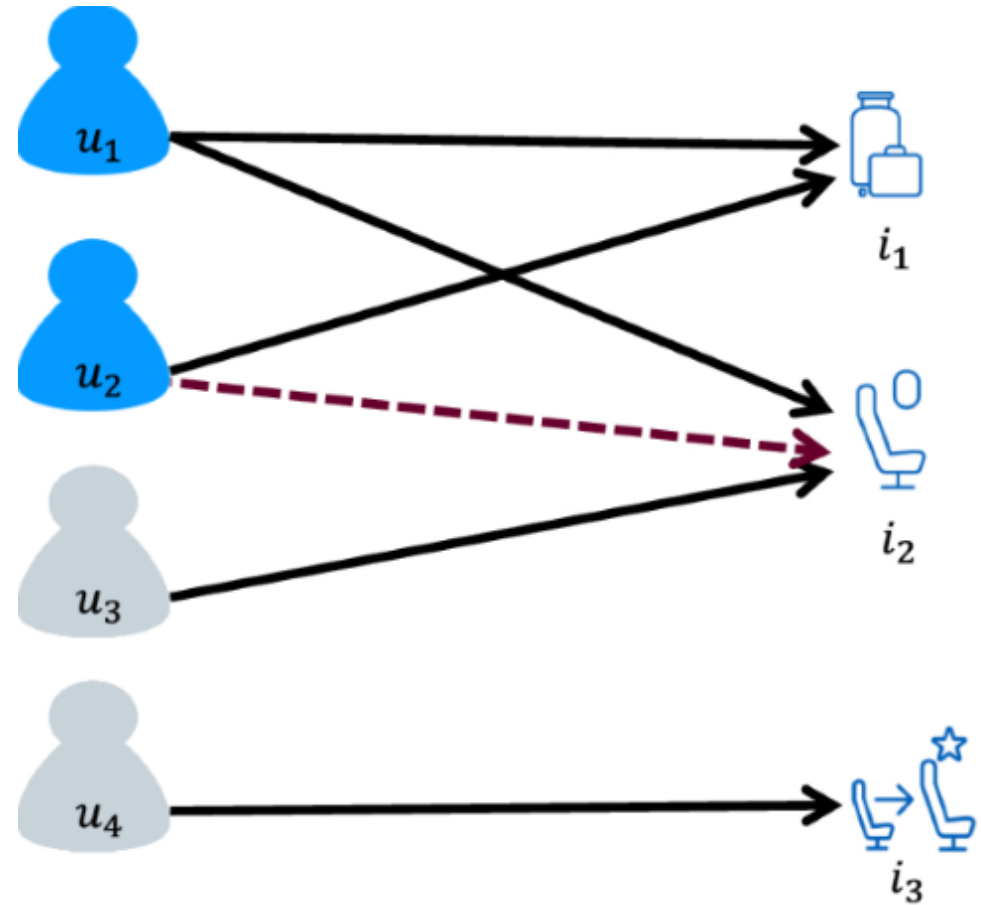
df_i = number of documents containing i

N = total number of documents

Якщо слово зустрічається в меншій кількості документів, і більше в даному документі, то його вага буде більшою

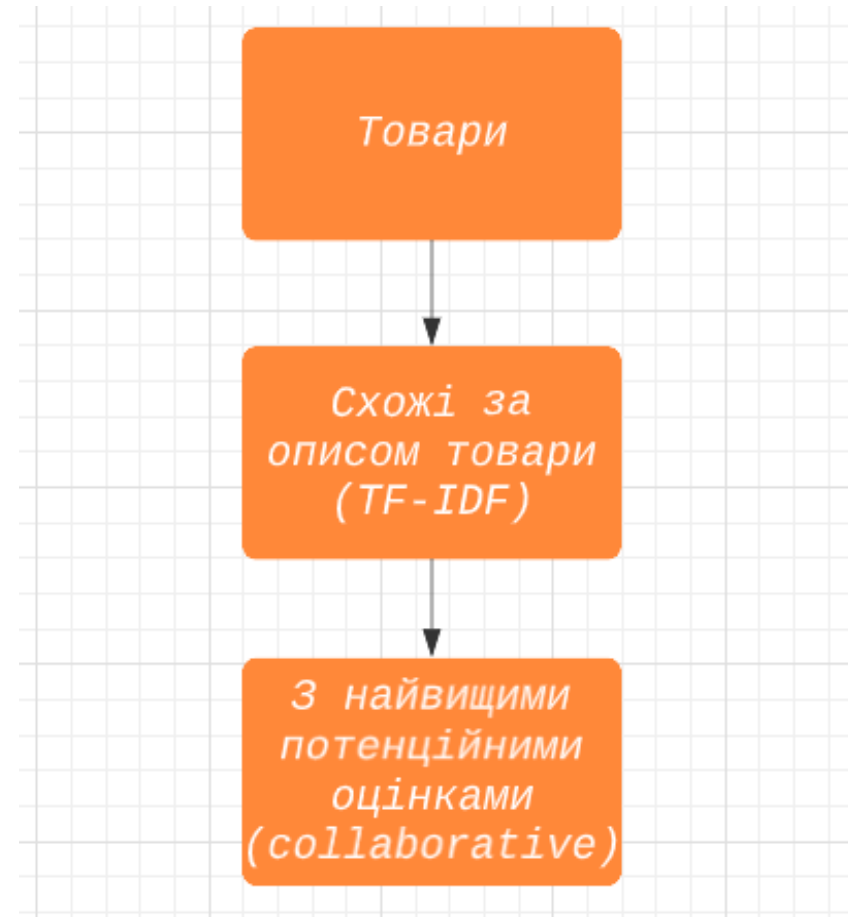
Передбачення оцінок (collaborative)

За допомогою weighted-mean розраховуємо оцінки найбільш подібних товарів

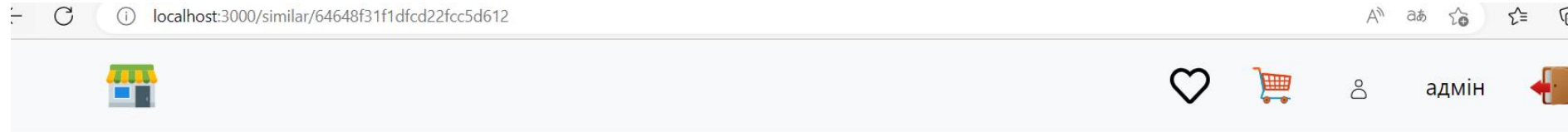


Функція рекомендації

1. Знаходимо k схожих за описом продуктів (content-based)
2. Передбачуємо, як користувач оцінить продукти, отримані в пункті 1 (collaborative)
3. Обираємо n з найкращими оцінками з отриманих в пункті 2



Робота системи



ghoul hood

це легендарний токійський
Гуль. Дізнайся ...

650\$



to cart



ball z pants

Штани мають принт з
образами улюблених п ...

720\$



to cart



Замок футболка

Має принт замок з аніме Хаяо
Міядзакі "З ...

565\$



to cart



manga pants

Штани мають принт з
зображенням персонаж ...

720\$



to cart

API рекомендаційної системи



Flask

Висновок

- Досліджено актуальність поставленої задачі
- Досліджено підходи до розробки веб-додатків
- Розроблено інтернет-магазин для продажу товарів пов'язаних з аніме
- Досліджені різні види рекомендаційних систем
- Створено гібридну багатоскладову рекомендаційну систему для онлайн-магазину

Дякую за увагу !!!