Advanced Recommendation System for an E-commerce Platform

- **مقدمه**

این سند طراحی و پیاده‌سازی یک سیستم پیشرفته توصیه‌گر برای یک پلتفرم تجارت الکترونیک را توضیح می‌دهد. این سیستم محصولات را بر اساس تاریخچه مرور و خرید کاربران، رفتار کاربران مشابه و سیگنال‌های زمینه‌ای مانند زمان روز، فصلی بودن و نوع دستگاه به آن‌ها پیشنهاد می‌دهد. این سیستم به گونه‌ای طراحی شده است که بتواند یک کاتالوگ بزرگ از محصولات و کاربران را مدیریت کند و در عین حال تعادل بین مرتبط بودن و تنوع در توصیه‌ها را حفظ نماید.

- **ساختارهای داده**

انتخاب ساختارهای داده در سیستم توصیه‌گر برای اطمینان از ذخیره‌سازی، بازیابی و پردازش کارآمد داده‌ها بسیار حیاتی است. در ادامه، دلیل انتخاب هر ساختار داده، چالش‌های مربوطه و علت انتخاب این ساختارها توضیح داده شده است.

- **اطلاعات کاربران**

ساختار:

```
users = {
  1: {"name": "Alice", "location": "New York", "device": "mobile"},
  2: {"name": "Bob", "location": "Los Angeles", "device": "desktop"},
}
```

هدف:

- ذخیره اطلاعات کاربران شامل شناسه کاربری، نام، موقعیت جغرافیایی و نوع دستگاه.
- استفاده برای شخصی‌سازی توصیه‌ها بر اساس ترجیحات کاربر، موقعیت مکانی و سازگاری با دستگاه.

هر کاربر به‌صورت یک دیکشنری نمایش داده می‌شود که امکان دسترسی سریع به ویژگی‌های کاربر را فراهم می‌کند. این لیست انعطاف‌پذیر است تا ویژگی‌های بیشتری را در آینده ذخیره کند. همچنین فیلد device برای توصیه‌های مبتنی بر دستگاه ضروری است و اطمینان حاصل می‌کند که محصولات توصیه‌شده با دستگاه کاربر سازگار هستند.

- **اطلاعات محصولات**

ساختار:

```
products = {
```

```
    101: {"name": "Wireless Earbuds", "category": "Electronics", "tags": ["audio", "wireless", "Bluetooth"], "rating": 4.5, "device_suitability": ["mobile", "tablet"]},

    102: {"name": "Smartphone Case", "category": "Accessories", "tags": ["phone", "protection", "case"], "rating": 4.2, "device_suitability": ["mobile"]},

}
```

هدف:

- ذخیره اطلاعات محصولات از جمله شناسه، نام، دسته‌بندی، برچسب‌ها و امتیاز کاربران و دستگاه های قابل انطباق.
- استفاده برای فیلترگذاری مبتنی بر محتوا، شناسایی محبوبیت محصول و ایجاد توصیه‌ها بر اساس ویژگی‌های محصول.

فیلدهای `tags` و `category` امکان توصیه‌ی محصولات مشابه بر اساس تعاملات کاربر را فراهم می‌کنند. این ساختار به‌راحتی قابل توسعه برای اضافه کردن اطلاعات اضافی مانند قیمت، برند و میزان موجودی است.

- **تاریخچه مرور کاربران**

ساختار:

```
browsing_history = {

  1: [(101, "2025-03-04 10:00:00"), (103, "2023-10-01 10:05:00")],

  2: [(102, "2025-03-04 11:30:00")],

}
```

هدف:

- ثبت محصولات مشاهده‌شده توسط کاربران همراه با تاریخ و زمان هر تعامل.
- تحلیل علاقه‌مندی‌های کاربران حتی در صورت عدم خرید.

ذخیره سازی به صورت دیکشنری با کلید های آیدی کاربران و مقادیر (product_id, timestamp) فیلد timestamp امکان تحلیل الگوهای مرور کاربران، مانند ساعات اوج فعالیت یا روندهای فصلی را فراهم می‌کند. ترکیب user_id و product_id برای ساخت پروفایل کاربران و استفاده در فیلترگذاری مشارکتی مفید است.

- **تاریخچه خرید کاربران**

ساختار:

```
purchase_history = {
  1: [(101, 1, "2025-03-04 10:00:00")],
  2: [(105, 2, "2025-03-04 12:00:00")],
}
```

هدف:

- ثبت محصولات خریداری‌شده توسط کاربران به همراه زمان خرید و تعداد اقلام.
- تحلیل علاقه‌مندی‌های کاربران، محصولات محبوب و روندهای خرید.

ذخیره سازی به صورت (product_id, quantity, timestamp) فیلد quantity در تحلیل الگوهای خرید (خرید عمده در مقابل خرید تکی) کمک می‌کند. امکان شناسایی روندهای خرید در بازه‌های زمانی مختلف را فراهم می‌کند.

- **سیگنال‌های زمینه‌ای**

ساختار:

```
contextual_signals = {
  "Electronics": {"peak_days": ["Friday", "Saturday"], "season": "Holiday"},
  "Fitness": {"peak_days": ["Monday", "Wednesday"], "season": "Summer"},
}
```

هدف:

- ثبت سیگنال‌های محیطی مانند روزهای اوج خرید، فصل‌های پررونق و نوع دستگاه.
- ارائه‌ی توصیه‌های متناسب با زمینه، مانند پیشنهاد محصولات تناسب‌اندام در تابستان.

فیلدهای peak_days و season به توصیه‌ی محصولات مرتبط با شرایط زمانی و محیطی کمک می‌کنند. و دسته‌بندی محصولات به شناسایی روندهای محبوبیت محصولات بر اساس سیگنال‌های زمینه‌ای کمک می کند.

- **الگوریتم طراحی**

سیستم توصیه‌گر از یک رویکرد ترکیبی استفاده می‌کند که چندین تکنیک توصیه را برای ارائه پیشنهادهای شخصی‌سازی‌شده، متنوع و مبتنی بر زمینه ترکیب می‌کند. در ادامه، هر الگوریتم را به‌طور دقیق توضیح می‌دهیم، دلایل انتخاب آن‌ها را بیان می‌کنیم و نحوه حل چالش‌های مرتبط با مرتبط بودن، تنوع و مقیاس‌پذیری را بررسی می‌کنیم.

- **تجزیه ماتریس (Matrix factorization)**

پیشنهاد محصولات بر اساس رفتار کاربران مشابه. این روش الگوهای تعامل کاربران با آیتم‌ها را شناسایی می‌کند تا پیش‌بینی کند که یک کاربر ممکن است چه محصولاتی را دوست داشته باشد، با توجه به اینکه کاربران مشابه چه محصولاتی را پسندیده‌اند.

نحوه عملکرد:

- **ماتریس تعامل کاربر-محصول**: یک ماتریس ایجاد می‌شود که در آن ردیف‌ها نشان‌دهنده کاربران، ستون‌ها نشان‌دهنده محصولات و مقادیر نشان‌دهنده تعاملات کاربران با محصولات (مانند خریدها یا بازدیدها) هستند.
- **تجزیه ماتریس**: این ماتریس با استفاده از تجزیه مقدار منفرد (SVD) به دو ماتریس با ابعاد پایین‌تر تبدیل می‌شود:
  - **بردارهای کاربر**: ویژگی‌های نهفته که ترجیحات کاربران را نمایش می‌دهند.
  - **بردارهای محصول**: ویژگی‌های نهفته که خصوصیات محصولات را نمایش می‌دهند.
- **تولید توصیه‌ها**: حاصل‌ضرب داخلی بردارهای کاربران و بردارهای محصولات محاسبه می‌شود تا احتمال تعامل یک کاربر با یک محصول مشخص شود. محصولاتی که بالاترین امتیاز را دارند به کاربر پیشنهاد می‌شوند.

چرا این روش انتخاب شده است؟

- **شناسایی تعاملات کاربر-محصول**: این روش به‌طور مؤثری الگوهای پنهان در رفتار کاربران را شناسایی می‌کند، حتی در صورتی که ترجیحات صریح کاربران در دسترس نباشد.
- **مقیاس‌پذیری**: در مجموعه‌داده‌های بزرگ به خوبی کار می‌کند، زیرا تجزیه ماتریس ابعاد داده‌ها را کاهش می‌دهد.
- **شخصی‌سازی**: بر اساس رفتار کاربران مشابه، پیشنهادهای شخصی‌سازی‌شده ارائه می‌دهد.

چالش‌ها:

- **Cold start**: برای کاربران یا محصولات جدید که هیچ سابقه تعاملی ندارند، عملکرد ضعیفی دارد.
- **پراکندگی داده‌ها**: ماتریس تعامل کاربر-محصول اغلب بسیار پراکنده است، که یافتن الگوهای معنادار را دشوار می‌کند.

راه‌حل‌ها:

- ترکیب این روش با فیلترینگ مبتنی بر محتوا و پیشنهاد محصولات محبوب برای مدیریت cold start.
- استفاده از نمایش‌های ماتریس پراکنده برای پردازش بهینه داده‌های پراکنده.

- **فیلترینگ مبتنی بر محتوا (Content-based filtering)**

پیشنهاد محصولات بر اساس شباهت بین پروفایل کاربران و ویژگی‌های محصولات. این روش بر ویژگی‌های محصولات تمرکز دارد که یک کاربر قبلاً با آن‌ها تعامل داشته است.

نحوه عملکرد:

- **بردارهای ویژگی محصول**: هر محصول بر اساس اطلاعات متا داده‌ای خود (مانند برچسب‌ها و دسته‌بندی‌ها) به‌عنوان یک بردار ویژگی نمایش داده می‌شود.
- **پروفایل کاربر**: پروفایل کاربر با تجمیع بردارهای ویژگی محصولاتی که او با آن‌ها تعامل داشته است (مانند بازدید یا خرید) ایجاد می‌شود.
- **تولید توصیه‌ها**: سیستم از جستجوی نزدیک‌ترین همسایه برای یافتن محصولاتی استفاده می‌کند که بردار ویژگی آن‌ها بیشترین شباهت را با پروفایل کاربر دارد. محصولاتی که بیشترین شباهت را دارند توصیه می‌شوند.

چرا این روش انتخاب شده است؟

- **مدیریت cold start**: برای کاربران یا محصولات جدید که سابقه تعامل محدودی دارند مفید است، زیرا به جای رفتار کاربران، بر ویژگی‌های محصولات تکیه دارد.
- **شخصی‌سازی**: توصیه‌های بسیار شخصی‌سازی‌شده‌ای را ارائه می‌دهد که بر اساس تعاملات قبلی کاربر با محصولات است.
- **شفافیت**: این روش توضیح‌پذیر است، زیرا توصیه‌ها بر اساس ویژگی‌های محصولات انجام می‌شوند.

چالش‌ها:

- **کیفیت متاداده**: اثربخشی این الگوریتم به کیفیت اطلاعات متا داده‌ای محصولات بستگی دارد.
- **محدودیت در تنوع توصیه‌ها**: ممکن است محصولات بسیار مشابهی را توصیه کند که کاربر قبلاً با آن‌ها تعامل داشته است.

راه‌حل‌ها:

- ترکیب این روش با فیلترینگ مشارکتی و پیشنهاد محصولات محبوب برای افزایش تنوع توصیه‌ها.
- استفاده از وزن دهی ترکیبی برای ایجاد تعادل میان فیلترینگ مبتنی بر محتوا و فیلترینگ مشارکتی.

- **محصولات محبوب**

پیشنهاد محصولاتی که بر اساس فراوانی خرید، امتیازات کاربران و سیگنال‌های زمینه‌ای محبوب یا trend هستند.

نحوه عملکرد:

- **امتیاز محبوبیت**: ترکیبی وزنی از فراوانی خرید و امتیازات کاربران برای محاسبه امتیاز محبوبیت هر محصول استفاده می‌شود.
- **محصولات trend**: سیگنال‌های زمینه‌ای (مانند فصلی بودن، روزهای اوج خرید) برای شناسایی دسته‌بندی‌های محصولاتی که trend هستند، مورد استفاده قرار می‌گیرند.

- **تولید توصیه‌ها**: محصولات محبوب و trend توصیه می‌شوند، به‌ویژه برای کاربران جدید یا در مواردی که داده‌های شخصی‌سازی‌شده محدود است.

چرا این روش انتخاب شده است؟

- **مدیریت cold start**: توصیه‌هایی را برای کاربران یا محصولات جدید بدون سابقه تعامل ارائه می‌دهد.
- **سادگی**: پیاده‌سازی آسان و مؤثر برای افزایش فروش محصولات محبوب.
- **تنوع**: با توصیه محصولات محبوب در بین تمامی کاربران، تنوع را افزایش می‌دهد.

چالش‌ها:

- **عدم شخصی‌سازی**: ممکن است توصیه‌ها با ترجیحات فردی کاربران مطابقت نداشته باشند.
- **نمایش بیش‌ازحد محصولات محبوب**: محصولات محبوب ممکن است توصیه‌ها را تحت تسلط قرار دهند و نمایش محصولات خاص‌تر را کاهش دهند.

راه‌حل‌ها:

- استفاده از توصیه‌های ترکیبی وزنی برای ایجاد تعادل بین محصولات محبوب و توصیه‌های شخصی‌سازی‌شده.
- اعمال محدودیت‌های تنوع برای اطمینان از ترکیب محصولات محبوب و خاص.

- **توصیه‌های مبتنی بر زمان**

پیشنهاد محصولات بر اساس روز هفته و روندهای فصلی.

نحوه عملکرد:

- **سیگنال‌های زمینه‌ای**: سیستم از داده‌های زمینه‌ای (مانند فصل، روزهای اوج خرید) برای شناسایی دسته‌بندی‌های محصولات پرروند استفاده می‌کند.
- **تولید توصیه‌ها**: محصولات از دسته‌های محبوب بر اساس زمینه زمانی فعلی پیشنهاد می‌شوند (مانند محصولات تناسب اندام در تابستان، لوازم الکترونیکی در تعطیلات).

چرا این روش انتخاب شده است؟

- **توصیه‌های مبتنی بر زمینه**: پیشنهادهایی ارائه می‌دهد که با زمان و فصل جاری مرتبط هستند.
- **افزایش ارتباط توصیه‌ها**: به شناسایی روندهای فصلی و ترجیحات زمانی کاربران کمک می‌کند.

چالش‌ها:

- **زمینه‌های پویا**: سیگنال‌های زمینه‌ای به‌طور مداوم تغییر می‌کنند و نیاز به بروزرسانی‌های بلادرنگ دارند.
- **دقت داده‌ها**: به داده‌های زمینه‌ای دقیق و به‌روز متکی است.

راه‌حل‌ها:

- استفاده از فیدهای داده‌ای بلادرنگ برای به‌روزرسانی پویا سیگنال‌های زمینه‌ای.
- ترکیب با فیلترینگ مشارکتی و فیلترینگ مبتنی بر محتوا برای بهبود شخصی‌سازی.

- **توصیه‌های مبتنی بر دستگاه**

پیشنهاد محصولات بر اساس نوع دستگاه کاربر (مانند موبایل، دسکتاپ).

نحوه عملکرد:

- **تناسب با دستگاه**: محصولات بر اساس تناسب آن‌ها با دستگاه کاربر فیلتر می‌شوند (مثلاً محصولات سازگار با موبایل برای کاربران موبایلی).
- **تولید توصیه‌ها**: محصولاتی که با دستگاه کاربر سازگار هستند توصیه می‌شوند.

چرا این روش انتخاب شده است؟

- **سازگاری با دستگاه**: اطمینان از اینکه توصیه‌ها با دستگاه کاربر سازگار هستند و تجربه کاربری بهتری فراهم می‌شود.
- **شخصی‌سازی بیشتر**: یک لایه اضافی از شخصی‌سازی را بر اساس نوع دستگاه کاربر اضافه می‌کند.

چالش‌ها:

- **داده‌های محدود**: به اطلاعات دقیق درباره تناسب محصولات با دستگاه‌های مختلف نیاز دارد.
- **تخصصی شدن بیش‌ازحد**: ممکن است توصیه‌ها را بیش‌ازحد محدود کند و فقط به گروه کوچکی از محصولات متمرکز شود.

راه‌حل‌ها:

- استفاده از توصیه‌های ترکیبی برای ایجاد تعادل بین توصیه‌های مبتنی بر دستگاه و سایر عوامل (مانند محبوبیت، فیلترینگ مبتنی بر محتوا).
- اطمینان از اینکه اطلاعات تناسب با دستگاه جامع و به‌روز هستند.

- **تکنیک‌های بهینه‌سازی**

- **caching و پیش‌محاسبه**

کاهش تاخیر و بهبود عملکرد از طریق کش کردن داده‌هایی که به‌طور مکرر مورد استفاده قرار می‌گیرند.

پیاده‌سازی:

توصیه‌ها، بردارهای ویژگی محصولات و فاکتورهای SVD با استفاده از Redis کش می‌شوند. مدت زمان اعتبار (TTL) برای توصیه‌ها یک ساعت و برای داده‌های پیش‌محاسبه‌شده ۲۴ ساعت تنظیم شده است.

چرا این روش انتخاب شده است؟

بهبود زمان پاسخ و کاهش بار روی موتور توصیه‌گر.

- **پردازش موازی**

مدیریت مجموعه‌داده‌های بزرگ به‌طور کارآمد از طریق موازی‌سازی وظایف پردازشی.

پیاده‌سازی:

بردارهای ویژگی محصولات و پروفایل‌های کاربران به‌طور موازی با استفاده از **ProcessPoolExecutor** محاسبه می‌شوند.

چرا این روش انتخاب شده است؟

افزایش سرعت پردازش داده‌های حجیم و تضمین مقیاس‌پذیری.

- **نمایش ماتریس پراکنده**

مدیریت کارآمد ماتریس تعامل کاربر-محصول برای کاهش مصرف حافظه و افزایش کارایی محاسباتی برای داده‌های پراکنده در مجموعه‌داده‌های بزرگ.

پیاده‌سازی:

ماتریس تعامل کاربر-محصول به‌صورت **ماتریس پراکنده** با استفاده از **scipy.sparse.csr_matrix** نمایش داده می‌شود.

- **موازنه‌ها و چالش‌ها**

- **تنوع**

ایجاد تعادل بین توصیه‌های شخصی‌سازی‌شده و نمایش محصولات جدید (تنوع).

راه‌حل:

استفاده از رویکرد ترکیبی که شامل **فیلترینگ مشارکتی، فیلترینگ مبتنی بر محتوا و توصیه محصولات محبوب** است تا هم ارتباط و هم تنوع را بهبود دهد.

- **Cold start**

کاربران یا محصولات جدید که سابقه تعامل محدودی دارند، چالش‌هایی برای توصیه‌های شخصی‌سازی‌شده ایجاد می‌کنند.

راه‌حل:

استفاده از **فیلترینگ مبتنی بر محتوا و توصیه محصولات محبوب**.

- **مقیاس‌پذیری**

سیستم باید بتواند مجموعه‌داده‌های بزرگ شامل کاربران و محصولات فراوان را به‌طور کارآمد مدیریت کند.

راه‌حل:

استفاده از **کشینگ، پردازش موازی و نمایش ماتریس پراکنده** برای بهینه‌سازی عملکرد.

- **پیاده‌سازی**

- **ماژول اصلی**: main.py فرآیند توصیه را مدیریت کرده و نتایج توصیه‌های مختلف را ترکیب می‌کند.
- **بارگذاری داده‌ها**: data_loading.py وظیفه بارگذاری و ایندکس کردن داده‌های کاربران و محصولات را بر عهده دارد.
- **مهندسی ویژگی**: feature_engineering.py بردارهای ویژگی محصولات و پروفایل‌های کاربران را محاسبه می‌کند.
- **فاکتورگیری ماتریسی**: matrix_factorization.py عملیات تجزیه مقادیر منفرد (SVD) را برای تولید فاکتورهای کاربران و محصولات انجام می‌دهد.
- **الگوریتم‌های توصیه‌گر**: recommendation_algorithms.py پیاده‌سازی الگوریتم‌های مختلف توصیه را بر عهده دارد.
- **پروفایل‌های کاربران**: user_profiles.py پروفایل‌های کاربران را بر اساس تاریخچه مرور و خرید آن‌ها محاسبه می‌کند.
- **پیکربندی سیستم**: setup.py تنظیمات مربوط به Redis و لاگ‌گیری را انجام می‌دهد.