سیستم های توسعه گر مبتنی برکاربر و مبتنی بر محصول

زهرا داوردوست

دانشگاه تبریز

نام درس : داده کاوی

مقدمه

سیستم‌های توصیه‌گر، موتورهای نرم‌افزاری‌ای هستند که به منظور پیشنهاد آیتم‌ها به کاربران بر اساس علاقه‌ها، تعاملات و تجربیات قبلی طراحی شده‌اند. این موتورها با ارائه پیشنهادهای شخصی‌سازی‌شده از فیلم‌ها، نمایش‌های تلویزیونی، محصولات دیجیتال، کتاب‌ها، مقالات و خدمات مختلف، به کاربران کمک می‌کنند تا آیتم‌های مورد علاقه خود را پیدا کنند. همچنین این سیستم‌ها به کسب‌وکارها کمک می‌کنند تا فروش خود را افزایش دهند و تجربه کاربری بهتری برای مشتریان ایجاد کنند. به عنوان مثال، آمازون میلیون‌ها محصول در وب‌سایت خود دارد که کاربران ممکن است در پیدا کردن و انتخاب محصولات مورد نظر دچار مشکل شوند. با استفاده از سیستم‌های توصیه‌گر، کاربران می‌توانند به راحتی محصولات مورد نیاز خود را پیدا کنند، تجربه کاربری بهتری داشته باشند و به استفاده مستمر از سایت ترغیب شوند.

مزایای استفاده از سیستم‌های توصیه عبارتند از:

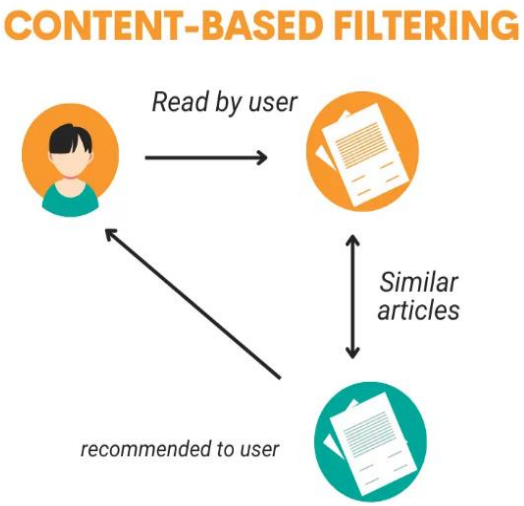
- افزایش فروش: اصلی‌ترین دلیل سرمایه‌گذاری شرکت‌ها در این سیستم‌ها، افزایش درآمد است. استفاده از سیستم‌های توصیه منجر به افزایش فروش می‌شود و همچنین تعامل بیشتر مصرف‌کنندگان را در سایت فراهم می‌کند که منجر به زمان بیشتری صرف شده در سایت می‌شود.
- بار کمتر بر سیستم: با فیلتر کردن بهترین گزینه‌ها برای هر کاربر، سیستم‌های توصیه بهبود فروش را فراهم می‌کنند و در عین حال بار کاری کمتری را بر روی سیستم ایجاد می‌کنند که در نتیجه کاهش هزینه‌ها در طولانی مدت را ایجاد می‌کند.
- افزایش تعامل و رضایت: با ارائه مداوم محتواهای شخصی‌سازی شده به مشتریان، سیستم‌های توصیه باعث ادامه تعامل مشتریان با برنامه یا وب‌سایت می‌شوند. این سیستم‌ها تجربه کاربری را با بهینه‌سازی محتواها، بهبود می‌بخشند و رضایت مشتری از محتوای مرتبط را افزایش می‌دهند.

انواع سیستم های توصیه

بسته به محصولات یا خدماتی که یک کسب‌وکار ارائه می‌کند، ممکن است سیستم‌های پیشنهادی متفاوتی ایجاد شود. چند نمونه از سیستم های مختلف عبارتند از :

Content-Based Filtering

الگوریتم فیلترینگ محتوایی شباهت محصولات را بررسی می‌کند. سیستم توصیه محصولاتی با دسته‌بندی مشابه به کاربرانی که پیش‌تر با آن‌ها تعامل داشته‌اند، پیشنهاد می‌دهد.

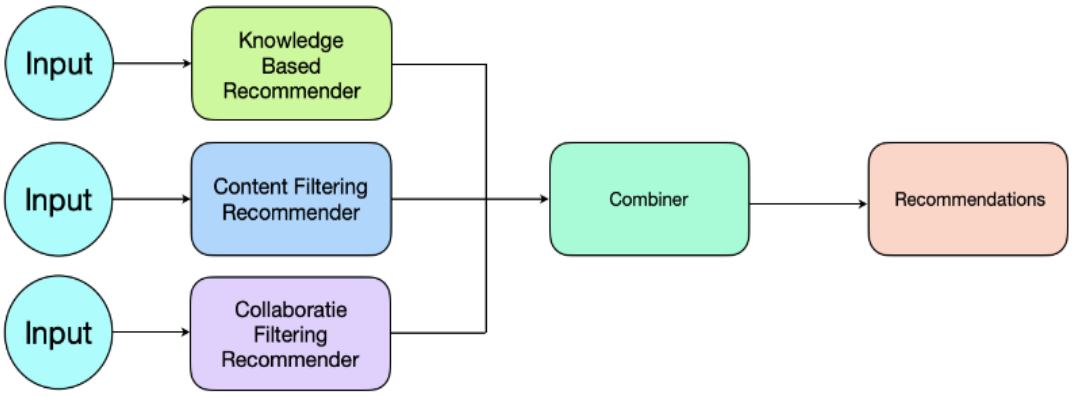


به عنوان مثال، اگر سه فیلم آخری که تماشا شده باشد، از ژانر کمدی باشند، سیستم سایر فیلم‌ها یا نمایش‌های کمدی مشابه را توصیه می‌کند. این توصیه‌ها برای محصولات مختلف می‌توانند از پردازش تصویر یا پردازش زبان طبیعی برای مطابقت با مواردی که در ظاهر، عنوان یا توصیف آنها شباهت دارند، استفاده کنند.

باید توجه داشت که توصیه‌های مبتنی بر شباهت ممکن است با مشکل شروع سرد مواجه شوند. مشکل شروع سرد زمانی پیش می‌آید که داده‌های کافی برای تحلیل وجود نداشته باشد. بنابراین، سیستم‌های توصیه که در ابتدای راه اجرا می‌شوند، قادر به ارائه گزینه‌های دقیق و عالی نیستند زیرا نیاز به جمع‌آوری و آموزش دارند.

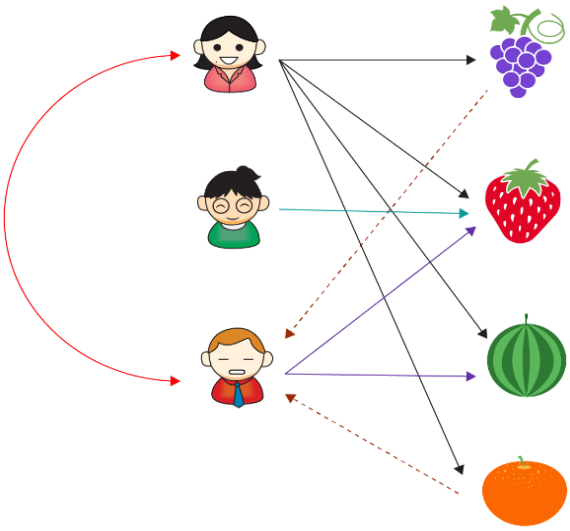
Hybrid Filtering

فیلتر ترکیبی از فیلتر مشارکتی و مبتنی بر محتوا استفاده می کند و از مزایای یکدیگر استفاده می کند. مطالعات متعددی که عملکرد سیستم‌های فیلتر هیبریدی را با سیستم‌های مشارکتی و محتوایی به تنهایی مقایسه کرده‌اند، نشان داده‌اند که سیستم‌های ترکیبی دقت بهتری دارند. ترکیب هر دو الگوریتم می تواند چندین مشکل مانند مشکل شروع سرد را حذف کند و به جمع آوری سریع داده ها کمک کند. بسیاری از سایت های مورد علاقه ما مانند گوگل، یوتیوب و نتفلیکس از فیلتر ترکیبی در سیستم های توصیه خود استفاده می کنند.



Collaborative Filtering

فیلتر مشارکتی تکنیک یا روشی برای پیش‌بینی سلیقه کاربر و یافتن مواردی است که کاربر ممکن است بر اساس اطلاعات جمع‌آوری‌شده از سایر کاربران با سلیقه یا ترجیحات مشابه ترجیح دهد. این واقعیت اساسی را در نظر می گیرد که اگر شخص X و شخص Y واکنش خاصی نسبت به برخی موارد داشته باشند، ممکن است برای موارد دیگر نیز همین نظر را داشته باشند.



دو نوع محبوب فیلتر مشارکتی عبارتند از:

Item-based

در اینجا، ما رابطه بین جفت آیتم ها را بررسی می کنیم کاربری که Y را خریداری کرده است، Zرا نیز خریداری کرده است . ما رتبه‌بندی گمشده را با کمک رتبه‌بندی‌هایی که کاربر به موارد دیگر داده است، پیدا می‌کنیم .بیایید در مورد فیلتر مشارکتی مبتنی بر آیتم با جزئیات صحبت کنیم. این فیلتر برای اولین بار توسط آمازون در سال ۱۹۹۸ اختراع و استفاده شد. به جای تطبیق کاربر با مشتریان مشابه، فیلتر اشتراکی مورد به مورد، هر یک از اقلام خریداری شده و رتبه‌بندی شده کاربر را با موارد مشابه مطابقت می‌دهد، سپس آن موارد مشابه را در فهرست توصیه‌ای ترکیب می‌کند. حال، اجازه دهید در مورد چگونگی کارکرد آن صحبت کنیم.

شباهت آیتم به آیتم: اولین قدم ساختن مدل با یافتن شباهت بین همه جفت‌های آیتم است. شباهت بین جفت اقلام را می توان به روش های مختلفی یافت. یکی از رایج ترین روش ها استفاده از تشابه کسینوس است.

فرمول تشابه کسینوس:

$$Similarity(\vec{A},\vec{B}) = \frac{\vec{A}\cdot\vec{B}}{||\vec{A}||*||\vec{B}||}$$

محاسبه پیش بینی: مرحله دوم شامل اجرای یک سیستم توصیه می شود. از مواردی استفاده می کند (که قبلاً توسط کاربر رتبه بندی شده است) که بیشترین شباهت به مورد گم شده را دارند برای ایجاد رتبه. از این رو سعی می کنیم بر اساس رتبه بندی محصولات مشابه، پیش بینی هایی ایجاد کنیم. ما این را با استفاده از فرمولی محاسبه می کنیم که رتبه بندی یک آیتم خاص را با استفاده از مجموع وزنی رتبه بندی سایر محصولات مشابه محاسبه می کند.

$$rating(U, I_i) = \frac{\sum_j rating(U, I_j) * s_{ij}}{\sum_j s_{ij}}$$

مثال:

اجازه دهید یک مثال را در نظر بگیریم. در زیر یک جدول مجموعه ای ارائه شده است که شامل برخی از موارد و کاربری است که به آن موارد امتیاز داده است. رتبه بندی صریح است و در مقیاس ۱ تا ۵ است. هر ورودی در جدول نشان دهنده رتبه ای است که کاربر **i** به **j** امین مورد داده است. در بیشتر موارد اکثر سلول ها خالی هستند زیرا کاربر فقط برای چند مورد امتیاز می دهد. در اینجا، ما ٤ کاربر و ۳ مورد را گرفته ایم. ما باید رتبه‌بندی‌های گمشده را برای کاربر مربوطه پیدا کنیم.

User/Item	Item_1	Item_2	Item_3
User_1	2	–	3
User_2	5	2	–
User_3	3	3	1
User_4	–	2	2

مرحله ۱: یافتن شباهت های همه جفت آیتم ها.

جفت آیتم ها را تشکیل دهید. برای مثال در این مثال، جفت‌های آیتم عبارتند از**Item_1)** ، **(Item_2، Item_1)**، **(Item_3**، و **Item_2)**، **.(Item_3)**هر مورد را برای جفت شدن یکی یکی انتخاب کنید. پس از این، همه کاربرانی را پیدا می کنیم که به هر دو مورد در جفت آیتم امتیاز داده اند. برای هر مورد یک بردار تشکیل دهید و شباهت بین دو مورد را با استفاده از فرمول کسینوس ذکر شده در بالا محاسبه کنید.

$$Similarity(I1, I2) = \frac{(5*2)+(3*3)}{\sqrt{5^2+3^2}\sqrt{2^2+3^2}} = 0.90$$

$$Similarity(I2, I3) = \frac{(3*1)+(2*2)}{\sqrt{3^2+2^2}\sqrt{1^2+2^2}} = 0.869$$

$$Similarity(I1, I3) = \frac{(2*3)+(3*1)}{\sqrt{2^2+3^2}\sqrt{3^2+1^2}} = 0.789$$

مرحله ۲: ایجاد رتبه بندی های گم شده در جدول

حال در این مرحله رتبه بندی هایی که در جدول وجود ندارد را محاسبه می کنیم.

$$r(U_1, I_2) = \frac{r(U_1,I_1)*s_{I_1I_2}+r(U_1,I_3)*s_{I_3I_2}}{s_{I_1I_2}+s_{I_3I_2}} = \frac{(2*0.9)+(3*0.869)}{(0.9+0.869)} = 2.49$$

$$r(U_2, I_3) = \frac{r(U_2,I_1)*s_{I_1I_3}+r(U_2,I_2)*s_{I_2I_3}}{s_{I_1I_3}+s_{I_2I_3}} = \frac{(5*0.789)+(2*0.869)}{(0.789+0.869)} = 3.43$$

$$r(U_4, I_1) = \frac{r(U_4,I_2)*s_{I_1I_2}+r(U_4,I_3)*s_{I_1I_3}}{s_{I_1I_2}+s_{I_1I_3}} = \frac{(2*0.9)+(2*0.789)}{(0.9+0.789)} = 2.0$$

برای پیاده سازی این الگوریتم از چندین مرحله پیروی میکنیم

مرحله ۰ :

الگوریتم توصیه فیلتر مشارکتی مبتنی بر آیتم

ابتدا، بیایید بفهمیم که فیلتر مشارکتی مبتنی بر آیتم چگونه کار می کند.

فیلتر مشارکتی مبتنی بر آیتم، توصیه هایی را بر اساس تعاملات کاربر-محصول در گذشته ارائه می دهد. فرض پشت سر این الگوریتم این است که کاربران محصولات مشابه را دوست دارند و محصولات مشابه را دوست ندارند، بنابراین به محصولات مشابه امتیازات مشابهی می دهند.

الگوریتم فیلتر مشارکتی مبتنی بر آیتم معمولاً مراحل زیر را دارد:

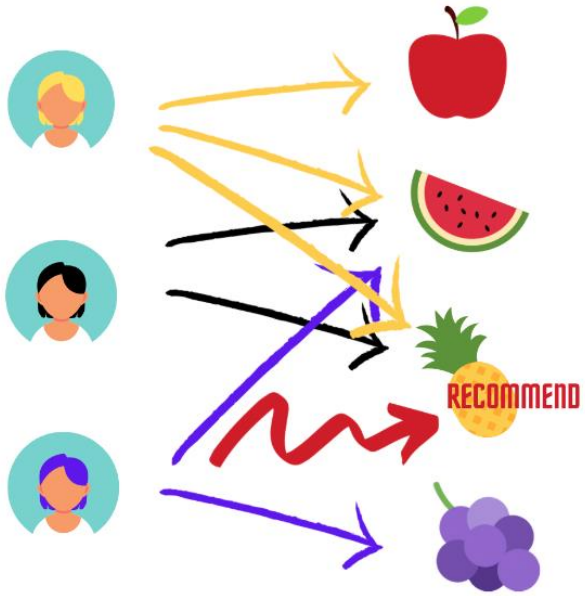
- محاسبه امتیاز شباهت آیتم‌ها بر اساس همه رتبه‌بندی‌های کاربران.
- شناسایی **n** مورد برتر که بیشترین شباهت را به آیتم مورد علاقه دارند.
- محاسبه میانگین وزنی امتیاز موارد مشابه توسط کاربر.
- رتبه‌بندی آیتم‌ها بر اساس امتیازات و انتخاب **n** مورد برتر برای توصیه.

این نمودار نحوه عملکرد فیلتر مشارکتی مبتنی بر آیتم را با یک مثال ساده توضیح می‌دهد:

خانم بلوند به سیب، هندوانه و آناناس علاقه دارد. خانم بلک به هندوانه و آناناس علاقه‌مند است. خانم ارغوان به هندوانه و انگور علاقه دارد.

به دلیل اینکه هندوانه و آناناس مورد علاقه مشترک افراد زیادی هستند، این دو مورد به عنوان اقلام مشابه در نظر گرفته می‌شوند.

از آنجا که خانم ارغوان به هندوانه علاقه‌مند است اما هنوز آناناس را امتحان نکرده است، سیستم توصیه آناناس را به او پیشنهاد می‌دهد.



مرحله ۱:

وارد کردن کتابخانه‌های پایتون

در این مرحله اول، کتابخانه‌های**pandas** ، **numpy**و **scipy.stats** را که برای پردازش داده‌ها و انجام محاسبات لازم هستند، وارد می‌کنیم.

همچنین، کتابخانه **seaborn** را برای تجسم داده‌ها و تابع **cosine_similarity** را برای محاسبه نمرات شباهت وارد می‌کنیم.

مرحله ۲:

دانلود و بارگذاری داده‌ها

در این آموزش از مجموعه داده **movielens** استفاده می‌شود که شامل رتبه‌بندی‌های واقعی کاربران برای فیلم‌ها است.

در مرحله ۲، مراحل زیر را برای دریافت مجموعه داده دنبال می‌کنیم:

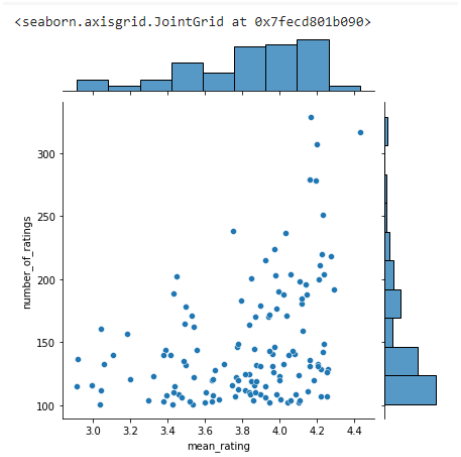
- به وبسایت **https://grouplens.org/datasets/movielens/** بروید.
- مجموعه داده ۱۰۰ **k**را با نام فایل **"ml-latest-small.zip"** دانلود کنید.
- فایل **"ml-latest-small.zip"** را از حالت فشرده خارج کنید.
- پوشه **"ml-latest-small"** را به پوشه پروژه خود کپی کنید.

چهار ستون در مجموعه داده‌های رتبه‌بندی، شناسه کاربر، شناسه فیلم، رتبه‌بندی و مهر زمانی وجود دارد.

این مجموعه داده بیش از ۱۰۰ هزار رکورد دارد و هیچ داده گمشده ای وجود ندارد.

رتبه بندی **k**۱۰۰ از ۶۱۰ کاربر در ۹۷۲۴ فیلم است. این رتبه ده مقدار منحصر به فرد از ۰.۵ تا ۵ دارد.

در مرحله بعدی بیایید داده های فیلم را بخوانیم تا نام فیلم ها را بدست آوریم.
مجموعه داده فیلم دارای شناسه فیلم، عنوان و ژانر است.
در مرحله بعد، بیایید داده های فیلم را میخوانیم تا نام فیلم ها را بدست آوریم.
مجموعه داده فیلم دارای شناسه فیلم، عنوان و ژانر است.
با استفاده از **movieID** به عنوان کلید تطبیق، اطلاعات فیلم را به مجموعه داده رتبه بندی اضافه کردیم و نام آن را **df** گذاشتیم.
بنابراین اکنون رتبه بندی فیلم و کاشی فیلم را در یک مجموعه داده داریم!



مرحله ۳:

تحلیل اکتشافی داده‌ها**(EDA)**

در مرحله ۳، باید فیلم‌ها را فیلتر کنیم و فقط آن‌هایی را که بیش از ۱۰۰ رتبه‌بندی دارند برای تحلیل نگه داریم. این کار به منظور مدیریت بهتر محاسبات توسط حافظه **Google Colab**انجام می‌شود.

برای انجام این کار، ابتدا فیلم‌ها را بر اساس عنوان گروه‌بندی کرده، تعداد رتبه‌بندی‌ها را می‌شماریم و فقط فیلم‌هایی که بیش از ۱۰۰ رتبه‌بندی دارند را نگه می‌داریم.
همچنین میانگین امتیاز فیلم‌ها را محاسبه می‌کنیم.
از خروجی **info**() متوجه می‌شویم که ۱۳۴ فیلم باقی مانده است.

مرحله ۴:

ماتریس فیلم کاربر ایجاد کنید

در مرحله ۴، مجموعه داده را به یک قالب ماتریسی تبدیل می کنیم. ردیف های ماتریس فیلم هستند و ستون های ماتریس کاربران هستند. مقدار ماتریس امتیاز کاربر فیلم در صورت وجود امتیاز است. در غیر این صورت، **"NaN"** را نشان می دهد.

userId	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	...
title																																									
2001: A Space Odyssey (1968)	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	3.0	NaN	NaN	NaN	4.0	NaN	NaN	NaN	2.0	3.0	5.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0	NaN	...
Ace Ventura: Pet Detective (1994)	NaN	NaN	NaN	NaN	3.0	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	2.5	2.0	NaN	NaN	NaN	NaN	NaN	NaN	3.0	NaN	NaN	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN	5.0	NaN	NaN	4.0	...
Aladdin (1992)	NaN	NaN	NaN	4.0	4.0	5.0	3.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN	3.0	NaN	NaN	3.5	3.0	5.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	5.0	NaN	NaN	NaN	NaN	3.0	4.0	4.0	...
Alien (1979)	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	4.0	NaN	4.0	4.0	NaN	1.5	NaN	4.0	NaN	NaN	NaN	NaN	3.5	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	...
Aliens (1986)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	NaN	NaN	3.0	NaN	2.0	NaN	3.0	NaN	NaN	NaN	1.0	4.0	NaN	3.0	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	...

5 rows × 597 columns

مرحله ۵:

نرمال سازی داده ها

در مرحله ۵، با کم کردن میانگین امتیاز هر فیلم، داده ها را عادی می کنیم. شباهت کسینوس محاسبه شده بر اساس داده های نرمال شده، شباهت کسینوس میانگین محور نامیده می شود.
پس از نرمال سازی، رتبه های کمتر از میانگین امتیاز فیلم یک مقدار منفی و امتیازات بیشتر از میانگین امتیاز فیلم یک مقدار مثبت دریافت می کنند.

مرحله ۶:

امتیاز شباهت را محاسبه کنید

روش های مختلفی برای اندازه گیری شباهت ها وجود دارد. همبستگی پیرسون و شباهت کسینوس دو روش پرکاربرد هستند.
در این آموزش ماتریس شباهت آیتم ها را با استفاده از همبستگی پیرسون محاسبه می کنیم.

مرحله ۷:

پیش بینی امتیاز کاربر برای یک فیلم

در مرحله ۷، امتیاز یک کاربر را برای یک فیلم پیش بینی می کنیم. بیایید از کاربر ۱ و فیلم **American Pie** به عنوان مثال استفاده کنیم.

پیش بینی روند زیر را دنبال می کند:

فهرستی از فیلم هایی که کاربر ۱ تماشا کرده و رتبه بندی کرده است ایجاد کنید.
شباهت های بین فیلم های رتبه بندی شده توسط کاربر ۱ و **American Pie** را رتبه بندی کنید.
n فیلم برتر با بیشترین امتیاز شباهت را انتخاب کنید.
رتبه بندی پیش بینی شده را با استفاده از میانگین وزنی نمرات شباهت و رتبه بندی های کاربر ۱ محاسبه کنید.

	title	rating	similarity_score
52	Mission: Impossible (1996)	-0.537037	0.510888
47	Twister (1996)	-0.321138	0.476518
16	Star Wars: Episode I - The Phantom Menace (1999)	0.892857	0.443614
10	Fugitive, The (1993)	1.007895	0.442128
19	Green Mile, The (1999)	0.851351	0.429560

مرحله ۸:

توصیه فیلم

در مرحله ۸، یک سیستم توصیه فیلم مورد-آیتم را در چهار مرحله ایجاد خواهیم کرد:
فهرستی از فیلم هایی که کاربر مورد نظر قبلاً آن را تماشا نکرده است ایجاد کنید.
فیلم تماشا نشده را مرور کنید و برای هر فیلم امتیازهای پیش بینی شده ایجاد کنید.

امتیاز پیش بینی شده فیلم تماشا نشده را از بالا به پایین رتبه بندی کنید.
k فیلم برتر را به عنوان توصیه برای کاربر مورد نظر انتخاب کنید.
تابع پایتون زیر چهار مرحله را پیاده سازی کرد. با ورودی **picked_userid**، **number_of_similar_item** و **number_of_recommendations**، می‌توانیم بهترین فیلم‌ها را برای کاربر و رتبه‌بندی‌های مربوط به آن‌ها دریافت کنیم.
توجه داشته باشید که رتبه‌بندی‌ها با استخراج میانگین امتیاز برای فیلم عادی می‌شوند، بنابراین اگر می‌خواهیم رتبه‌بندی‌های پیش‌بینی‌شده در همان مقیاس رتبه‌بندی اصلی باشد، باید مقدار میانگین را به رتبه‌بندی‌های پیش‌بینی‌شده اضافه کنیم.

```
[('Austin Powers: The Spy Who Shagged Me (1999)', 1.096288),
 ('Crouching Tiger, Hidden Dragon (Wo hu cang long) (2000)', 0.92924),
 ('Lord of the Rings: The Return of the King, The (2003)', 0.926824)]
```

User-based

فیلتر مشارکتی مبتنی بر کاربر یک تکنیک است که برای پیش‌بینی آیتم‌هایی که ممکن است مورد علاقه یک کاربر باشد، استفاده می‌شود. این پیش‌بینی بر اساس رتبه‌بندی‌هایی انجام می‌شود که سایر کاربرانی که سلیقه‌ای مشابه با کاربر هدف دارند، به آن آیتم‌ها داده‌اند. بسیاری از وب‌سایت‌ها از فیلتر مشارکتی برای ساخت سیستم‌های توصیه‌گر خود بهره می‌برند.
مراحل فیلتر مشارکتی مبتنی بر کاربر:
مرحله ۱: یافتن شباهت کاربران به کاربر هدف **U**. شباهت برای هر دو کاربر **"a"** و **"b"** را می توان از فرمول داده شده محاسبه کرد.
مرحله ۲: پیش‌بینی رتبه‌بندی از دست رفته یک آیتم در حال حاضر، کاربر مورد نظر ممکن است بسیار شبیه به برخی از کاربران باشد و ممکن است چندان شبیه به دیگران نباشد. از این رو، رتبه‌بندی‌هایی که به یک کالای خاص توسط کاربران مشابه‌تر داده می‌شود، باید وزن بیشتری نسبت به امتیازاتی که توسط کاربران کمتر مشابه و غیره داده می‌شود، داده شود. این مشکل با استفاده از روش میانگین وزنی قابل حل است. در این روش، امتیاز هر کاربر را با ضریب شباهت محاسبه شده با استفاده از فرمول ذکر شده در بالا ضرب می کنید. رتبه از دست رفته را می توان به صورت زیر محاسبه کرد.

$$Sim(a,b) = \frac{\sum_p (r_{ap} - \bar{r}_a)(r_{ab} - \bar{r}_b)}{\sqrt{\sum (r_{ap} - \bar{r}_a)^2} \sqrt{\sum (r_{bp} - \bar{r}_b)^2}}$$

$r_{up} : rating\ of\ user\ u\ against\ item\ p$
$p : items$

مثال: ماتریسی را در نظر بگیرید که رتبه چهار کاربر **Alice**، **U1**، **U2** و **U3** را در برنامه های خبری مختلف نشان می دهد. محدوده امتیاز از ۱ تا ۵ بر اساس محبوبیت کاربران از برنامه خبری است. "؟" نشان می دهد که کاربر به برنامه رتبه بندی نکرده است.

Name	Inshorts(I1)	HT(I2)	NYT(I3)	TOI(I4)	BBC(I5)
Alice	5	4	1	4	?
U1	3	1	2	3	3
U2	4	3	4	3	5
U3	3	3	1	5	4

مرحله ۱: محاسبه شباهت بین **Alice** و همه کاربران دیگر در ابتدا میانگین امتیازات همه کاربران به استثنای **I5** را محاسبه می کنیم زیرا توسط **Alice** رتبه بندی نشده است.

$$\bar{r}_{Alice} = 3.5$$
$$\bar{r}_{U1} = 2.25$$
$$\bar{r}_{U2} = 3.5$$
$$\bar{r}_{U3} = 3$$

ماتریس زیر به دست می آید:

average : $\bar{r}_i = \frac{\sum_p r_{ip}}{\sum p}$

new ratings: $r'_{ip} = r_{ip} - \bar{r}_i$

Name	Inshorts(I1)	HT(I2)	NYT(I3)	TOI(I4)
Alice	1.5	0.5	-2.5	0.5
U1	0.75	-1.25	-0.25	0.75
U2	0.5	-0.5	0.5	-0.5
U3	0	0	-2	2

اکنون شباهت آلیس و سایر کاربران را محاسبه می کنیم:

$$Sim(Alice, U1) = \frac{((1.5*0.75)+(0.5*-1.25)+(-2.5*-0.25)+(.5*0.75))}{\sqrt{(1.5^2+0.5^2+2.5^2+0.5^2)}\sqrt{(0.75^2+1.25^2+0.25^2+0.75^2)}} = 0.301$$

$$Sim(Alice, U3) = \frac{((1.5*0)+(0.5*0)+(-2.5*-2)+(.5*2))}{\sqrt{(1.5^2+0.5^2+2.5^2+0.5^2)}\sqrt{(0^2+0^2+2^2+2^2)}} = 0.707$$

مرحله ۲: پیش‌بینی رتبه‌بندی برنامه‌ای که توسط **Alice Now** رتبه‌بندی نشده است، رتبه **Alice** را برای برنامه خبری **BBC** پیش‌بینی می‌کنیم.

$$r_{(Alice,I5)} = \bar{r}_{Alice} + \frac{(sim(Alice,U1)*(r_{U1,I5}-\bar{r}_{U1}))+(sim(Alice,U2)*(r_{U2,I5}-\bar{r}_{U2}))+(sim(Alice,U3)*(r_{U3,I5}-\bar{r}_{U3})}{|sim(Alice,U1)|+|sim(Alice,U2)|+|sim(Alice,U3|}$$

پیاده سازی این مورد هم نسبتا شبیه به قبلی میباشد
تا مرحله ی ۵ مانند قبلی میباشد چرا ک **preprocess** ها انجام میشوند روی دیتا
مرحله ۶: شناسایی کاربران مشابه
روش های مختلفی برای اندازه گیری شباهت ها وجود دارد. همبستگی پیرسون و شباهت کسینوس دو روش پرکاربرد هستند.
در این آموزش ماتریس شباهت کاربر را با استفاده از همبستگی پیرسون محاسبه می کنیم.

```
# User similarity matrix using Pearson correlation
user_similarity = matrix_norm.T.corr()
user_similarity.head()
```

userId	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
userId																									
1	1.000000	NaN	NaN	0.391797	0.180151	-0.439941	-0.029894	0.464277	1.0	-0.037987	0.385758	NaN	NaN	0.175000	0.305392	0.293103	0.088608	0.386777	0.559040	NaN	0.029715	-0.944911	-0.286468	0.222515	-0.166667
2	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.000000	NaN	NaN	NaN	NaN	-0.752651	0.000000	-0.648886	-0.683130	NaN	NaN	0.612372	-0.772683	NaN	-0.912871	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	0.391797	NaN	NaN	1.000000	-0.394823	0.421927	0.704669	0.055442	NaN	0.360399	0.872872	NaN	NaN	0.215166	0.408228	0.046849	-0.194802	-0.071429	0.129813	0.405999	-0.076171	0.944911	-0.519675	0.150739	NaN
5	0.180151	NaN	NaN	-0.394823	1.000000	-0.006888	0.328889	0.030168	NaN	-0.777714	0.486854	1.0	NaN	0.252039	-0.012067	-0.684653	0.310835	0.159568	0.312031	0.000000	-0.339422	NaN	0.000000	0.193649	NaN

5 rows × 597 columns

مرحله ۷: استخر آیتم ها را باریک کنید
در مرحله ۷، با انجام کارهای زیر مجموعه آیتم ها را محدود می کنیم:

فیلم هایی که توسط کاربر مورد نظر تماشا شده است را حذف کنید (شناسه کاربری ۱ در این مثال).
فقط فیلم هایی را که کاربران مشابه تماشا کرده اند نگه دارید.
برای حذف فیلم‌های تماشا شده توسط کاربر هدف، فقط ردیف **userId=1** را در ماتریس کاربر-مورد نگه می‌داریم و مواردی را که مقادیر گمشده دارند حذف می‌کنیم.

title	Alien (1979)	American Beauty (1999)	American History X (1998)	Apocalypse Now (1979)	Back to the Future (1985)	Batman (1989)	Big Lebowski, The (1998)	Braveheart (1995)	Clear and Present Danger (1994)	Clerks (1994)	Clockwork Orange, A (1971)	Dances with Wolves (1990)	Dumb & Dumber (Dumb and Dumber) (1994)	Te
userId														
1	-0.392857	0.607143	0.607143	-0.392857	0.607143	-0.392857	0.607143	-0.392857	-0.392857	-1.392857	0.607143	-0.392857	0.607143	

برای اینکه فقط فیلم‌های کاربران مشابه باقی بماند، شناسه‌های کاربری را در ۱۰ لیست کاربر مشابه بالا نگه می‌داریم و فیلم را با تمام مقادیر از دست رفته حذف می‌کنیم. تمام مقادیر از دست رفته برای یک فیلم به این معنی است که هیچ یک از کاربران مشابه فیلم را تماشا نکرده اند.

```
# Movies that similar users watched. Remove movies that none of the similar users have watched
similar_user_movies = matrix_norm[matrix_norm.index.isin(similar_users.index)].dropna(axis=1, how='all')
similar_user_movies
```

title	Aladdin (1992)	Alien (1979)	Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)	Back to the Future (1985)	Batman Begins (2005)	Beautiful Mind, A (2001)	Beauty and the Beast (1991)	Blade Runner (1982)	Bourne Identity, The (2002)	Braveheart (1995)	Breakfast Club, The (1985)	Catch Me If You Can (2002)	Clerks (1994)	Clockwork Orange, A (1971)	Dark Knight, The (2008)	D
userId																
9	NaN	NaN	NaN	0.333333	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-0.666667	NaN	NaN	
108	NaN	NaN	0.466667	0.466667	NaN	0.466667	NaN	0.466667	NaN	NaN	-0.533333	0.466667	NaN	NaN	NaN	
154	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
366	NaN	NaN	NaN	NaN	-0.205882	NaN	NaN	NaN	NaN	-0.205882	NaN	NaN	NaN	NaN	-0.205882	
401	-0.382353	NaN	NaN	NaN	NaN	NaN	-0.382353	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
502	NaN	-0.375	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.125	NaN	
511	NaN	NaN	-0.653846	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
550	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-0.277778	NaN	NaN	-0.277778	
595	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
598	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.888889	NaN	NaN	NaN	NaN	NaN	NaN	

مرحله ۸: موارد را توصیه کنید

در مرحله ۸، تصمیم خواهیم گرفت که کدام فیلم را به کاربر مورد نظر توصیه کنیم. موارد توصیه شده با میانگین وزنی امتیاز شباهت کاربر و رتبه بندی فیلم تعیین می شود. رتبه بندی فیلم ها با نمرات شباهت وزن می شود، بنابراین کاربرانی که شباهت بالاتری دارند وزن های بالاتری دریافت می کنند.

این کد در میان آیتم ها و کاربران حلقه می زند تا امتیاز آیتم را به دست آورد، امتیاز را از بالا به پایین رتبه بندی می کند و ۱۰ فیلم برتر را برای توصیه به شناسه کاربر ۱ انتخاب می کند.

	movie	movie_score
16	Harry Potter and the Chamber of Secrets (2002)	1.888889
13	Eternal Sunshine of the Spotless Mind (2004)	1.888889
6	Bourne Identity, The (2002)	0.888889
29	Ocean's Eleven (2001)	0.888889
18	Inception (2010)	0.587491
3	Beautiful Mind, A (2001)	0.466667
5	Blade Runner (1982)	0.466667
12	Donnie Darko (2001)	0.466667
10	Departed, The (2006)	0.256727
31	Shawshank Redemption, The (1994)	0.222566

مرحله ۹: پیش بینی امتیازات

اگر هدف انتخاب موارد پیشنهادی است، داشتن رتبه آیتم ها کافی است. با این حال، اگر هدف پیش‌بینی امتیاز کاربر است، باید میانگین امتیاز امتیاز فیلم کاربر را دوباره به امتیاز فیلم اضافه کنیم.

```
# Calcuate the predicted rating
ranked_item_score['predicted_rating'] = ranked_item_score['movie_score'] + avg_rating

# Take a look at the data
ranked_item_score.head(m)
```

	movie	movie_score	predicted_rating
16	Harry Potter and the Chamber of Secrets (2002)	1.888889	6.281746
13	Eternal Sunshine of the Spotless Mind (2004)	1.888889	6.281746
6	Bourne Identity, The (2002)	0.888889	5.281746
29	Ocean's Eleven (2001)	0.888889	5.281746
18	Inception (2010)	0.587491	4.980348
3	Beautiful Mind, A (2001)	0.466667	4.859524
5	Blade Runner (1982)	0.466667	4.859524
12	Donnie Darko (2001)	0.466667	4.859524
10	Departed, The (2006)	0.256727	4.649584
31	Shawshank Redemption, The (1994)	0.222566	4.615423

چالش ها :

به دلیل سنگین بودن محاسبات و دردسترس نبودن منابع مجبور به استفاده از گوگل کولب شدم . ک سختی های اتصال و قطع نشدن اینترنت به شدت تایم طولانی ای گرفت.

در متن دیتاست متن هایی وجود داشت که به فرمت utf-8 نبودند و مجبور به پاک کردن آنها شدیم.